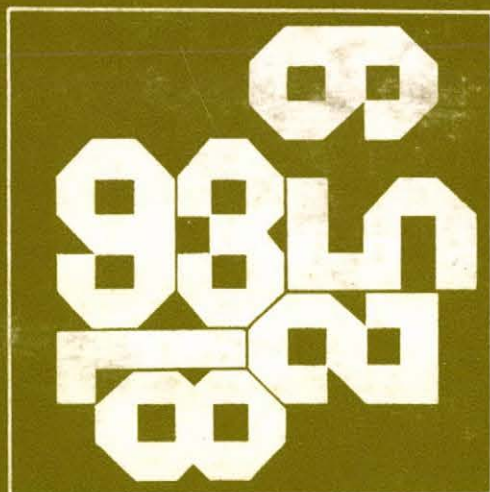


317.057  
1973  
tanulmányok

1/1973

MTA Számítástechnikai és Automatizálási Kutató Intézet Budapest





MÓDSZEREK BOOLE-FÜGGVÉNYEK MINIMÁLIS VAGY NEM REDUNDÁNS,  
 $\{\wedge, \vee, \neg\}$  VAGY  $\{\text{NOR}\}$  VAGY  $\{\text{NAND}\}$  BÁZISBELI, ZÁRÓJELES  
VAGY ZÁRÓJEL NÉLKÜLI FORMULÁINAK ELŐÁLLÍTÁSA

Pásztorné Varga Katalin

Tézisek

Budapest, 1973.



A kiadásért felelős:

Dr. Vámos Tibor

az

MTA Számítástechnikai és Automatizálási

Kutató Intézet

igazgatója

Készült az Országos Műszaki Könyvtár és Dokumentációs Központ  
házi sokszorosítójában  
F.v.: Janoch Gyula



## ELŐSZÓ

Jelen munka keletkezése jól meghatározott okokra vezethető vissza. Az egyre nagyobb teljesítményű számítógépek, bonyolult célgépek és elektronikus /mérő/ műszerek iránti igény a konstruktőrök feladatát egyre nehezíti. Viszont a már meglévő számítógépek is lehetővé teszik az újabb rendszerek tervezésénél az eddig alkalmazott módszerek számítógépre vitelét. A cél a tervezés minden fázisának megoldása számítógépen és egy egységes automatikus tervező programrendszer kifejlesztése. Jelenleg azonban az automata absztrakt strukturájának megtervezése szinte kizárólag a tervező feladata, mivel az automataelméletben eddig kidolgozott szintézis módszerek számítógépre vitele még nem nevezhető sikeresnek. Ezzel szemben a kialakított strukturát realizáló logikai hálózat tervezése, sőt legyártásának irányítása ma már a számítógépek hatáskörébe tartozik. A logikai hálózatok számítógépes tervezésének egyes fázisaiban többek között a matematikai logika ítéletkalkulusában elért eredményeket is felhasználják.

Az MTA Automatizálási Kutató Intézetben Uzsoky Miklós osztályvezető irányítása mellett a Digitális Áramkörök Szintézise és Realizációja Osztály egy csoportja foglalkozik a digitális áramkörök tervezésének különböző szakaszaira vonatkozó algoritmusok fejlesztésével, az ehhez kapcsolódó elméleti problémák megoldásával és az eredmények számítógépre vitelével.

Azok az ötletek és eredeti szempontok, amelyeket Uzsoky Miklós a módszerek realizáció szempontjából való megvitatása során vetett fel, nagymértékben hozzájárultak a szintézis feladat olyan megfogalmazásához, amely hatékony algoritmusok elérését tette lehetővé és elősegítette, hogy az eredmények a gyakorlat szempontjából értékesebbé váljanak. Hasznosnak bizonyultak számomra az osztályon dolgozó kollégákkal folytatott beszélgetések, különösen a minimalizálási feladat gyakorlati vonatkozásainak tisztázásában.

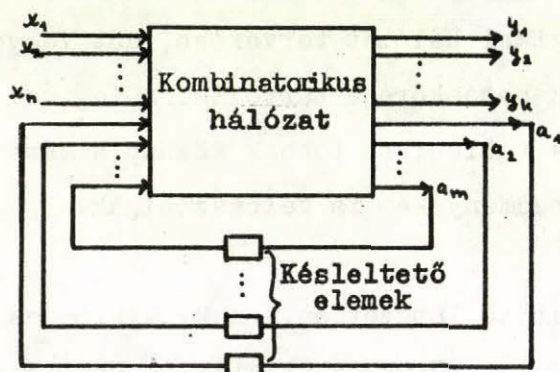
Köszönetemet fejezem ki aspiránsvezetőmnnek, Kalmár László akadémikusnak,



aki a disszertáció témáját állandóan figyelemmel kísérte. Megjegyzéseiért, értékes tanácsaiért és azért, hogy bátorításával hozzásegített a problémakör különböző szempontok szerinti megközelítéséhez, őszinte hálával tartozom.

Nagy segítséget jelentettek azok a baráti tanácsok és értékes észrevételek, amelyekkel dr. Fidirich Ilona kolléganőm segített a kézirat teljes átnézése során.

A kombinatorikus áramkörök sok esetben valamely realizálásra kerülő véges automatának képezik részét. A véges automaták logikai sémájának /hálózattal való ábrázolásának/ szokásos formája az 1. ábrán látható.



1. ábra

A kombinatorikus hálózat bizonyos elemi függvényeket realizáló funkcionális elemekből /pl. integrált áramkörökből/ felépített olyan hálózat, amely az  $x_1, x_2, \dots, x_n$  változók  $y_1, y_2, \dots, y_k, a_1, a_2, \dots, a_m$  Boole-függvényeit realizálja. Valamely kombinatorikus hálózat első közelítésben<sup>3</sup> az egyes függvényeket realizáló izolált kombinatorikus hálózatok halmazának tekinthető. Ez a magyarázata annak, hogy az elektronikus berendezések tervezése során felmerül a következő jól definiált probléma: adott Boole-függvény meghatározott bázisban való szintézisének automatizálása, figyelembe véve a szín-

<sup>3</sup> Ebben a dolgozatban a "közelítés", "első közelítés" és hasonló kifejezéseket a műszaki gyakorlatban használatos heurisztikus-empirikus értelemben értjük; pontos matematikai definíciójukra /hibabecslés segítségével/ nem törekszünk.



tézis eredményeként kapott formulát realizáló hálózatot alkotó alapelemek technológiai adottságait.

Valamely  $y_1, y_2, \dots, a_m$  kombinatorikus hálózat legyártásáig, logikai sémájának megadásától, egy sor munkafázis elvégzése szükséges. E dolgozatban a logikai séma kialakításának csak egyes problémáit érintjük. Összefoglaljuk a Boole-függvényekre vonatkozó, a disszertációban is felhasznált elméleti eredményeket, majd részletesen foglalkozunk a szintézis problémakörével.

A kombinatorikus hálózatok tervezésében a szintézis feladata a realizálandó Boole-függvény /vagy függvények/ megadása olyan formulával, amely a változókon és a bázisfüggvényeken kívül mást nem tartalmaz. Követelmény, hogy a formulát realizáló hálózat technológiailag megvalósítható, ezenkívül előállítási költsége a minimálisához közel legyen. Ennek megfelelően megköveteljük, hogy a szintézis eredményeként kapott formula valamilyen szempontból /pl. a változós szám, vagyis a változó előfordulások száma szerint/ minimális /optimális szintézis vagy minimalizálás/, vagy legalábbis irredundáns legyen /szintézis általában/.

A dolgozat egy részében foglalkozunk Boole-függvények  $\{\wedge, \vee, \neg\}$  bázisban diszjunktív normálformájú formuláinak változós szám szerinti minimalizálásával és irredundáns formulák felírásával. Áttekintést adunk az eddig ismert módszerekről, amelyek általában egyetlen teljesen meghatározott Boole-függvényre [2, 4, 19, 56, 57, 67] vagy  $\Phi$ -Boole-függvényre [27, 48], esetleg általános Boole-függvényre [17, 47] vonatkoznak. Foglalkozunk az egyes módszerek továbbfejlesztésével, amellyel e módszereknek számítógépes alkalmazás szempontjából való hatékonyságának lényeges javítását is sikerült elérnünk. Új módszereket dolgozunk ki a szintézis feladatra. Minden esetben olyan algoritmusokat adunk meg, amelyek a közepes teljesítményű számítógépek memóriakorlátait figyelembe veszik. Az algoritmusokat kiterjesztjük  $\Phi$ -Boole-függvények esetére is. Ezen túlmenően kidolgozunk általános Boole-függvények  $\{\wedge, \vee, \neg\}$  bázisban való minimalizálására ill. szintézisére alkalmas eljárást is. Jelen dolgozatban megadott egyik módszer kidolgozásával



Kalmár László professzor ur javaslatára kezdtem el foglalkozni. Az általa felvetett ötlet nyomán vizsgáltam, hogy a véges determinisztikus automaták állapotszám-minimalizálására kidolgozott módszerek alkalmazhatók-e a Boole-függvények minimalizálására ill. kapcsolatba hozhatók-e egymással e módszerek és a Boole-függvényekre vonatkozó szintézis módszerek /II.3./.

Nem kisebb jelentőségű, sőt a technikai fejlődés iránya miatt egyre fontosabb a klasszikus  $\{\wedge, \vee, \neg\}$  bázistól különböző bázisban való szintézis módszerek kidolgozása. E szintézis módszereknél a cél általában nem feltétlenül minimális, hanem csak irredundáns formula megadása. Ezen irredundáns formulákat realizáló hálózatokban szereplő, a bázisfüggvényeket realizáló funkcionális elemekre bizonyos technológiai korlátozások érvényesek. Ennek megfelelően olyan irredundáns formulákat kell megadni, amelyek a technológiai korlátozásoknak eleget tevő hálózatok realizálnak. Ezt az is indokolja, hogy általában a minimális alak nem tesz eleget a feltételeknek, ezenkívül a minimális formulák megkeresése összehasonlíthatatlanul időigényesebb feladat.

A dolgozatban kidolgozunk egy olyan szintézis módszert, amely  $\{\text{NAND}\}$  bázisban adja meg az irredundáns formulát. A szintetizálandó függvény lehet teljesen meghatározott Boole-függvény,  $\Phi$ -Boole-függvény vagy általános Boole-függvény. A szintézis módszer figyelembe veszi a bemenetszámra, a kimenetek terhelhetőségére és a hálózatmélységre vonatkozó korlátozásokat.

Befejezésül meg kell jegyezni, hogy a véges automata hálózatának fent említett alakja /1. ábra/, mint szekvenciális hálózat, idealizált. Ezért a kombinatorikus áramkörök szintézise csupán megközelítése a valóságos szekvenciális áramköri szintézisnek. Bár e terület elméletével számos kutató foglalkozik, szekvenciális áramkörökre vonatkozó számítógép-orientált szintézis módszerek alig ismertek. A valóságot jobban megközelítő szekvenciális áramköri szintézis méreteiben túlnő a közepes számítógépek keretein. Figyelembe véve a közeljövő számítógépeinek összehasonlíthatatlanul nagyobb teljesítményét, igen gyümölcsözőnek és hasznosnak látszik e kutatási terület művelése.



## BEVEZETÉS

A disszertáció tárgya Boole-függvényeket különböző bázisokban leíró, lehetőleg minimális formulákat megadó algoritmusok kidolgozása. A disszertációban Boole-függvényeket minimalizáló, már ismert eljárásokat általánosítunk  $\Phi$ -Boole-függvények esetére. A kidolgozott algoritmusok egy része nem követeli meg a függvény kitünttetett /más néven teljes/ diszjunktív normálformában /KDNF/ való megadását, sőt az eljárás kiinduló adataként megadott tetszőleges diszjunktív normálforma /DNF/ alakjában rejlő esetleges információt is felhasználják. Kidolgoztunk olyan algoritmusokat, amelyek a függvény KDNF-ja alapján teljesen meghatározott,  $\Phi$  - vagy általános Boole-függvényre  $\{\wedge, \vee, \neg\}$  bázisban irredundáns, esetleg minimális diszjunktív normálformát adnak. Megadunk olyan szintézis eljárást is, amely irredundáns formula felírását eredményezi  $\{\text{NAND}\}$  bázisban, meghatározott  $\Phi$  - vagy általános Boole-függvényekre. Ez az eljárás bizonyos korlátozó feltételeket is képes figyelembe venni.

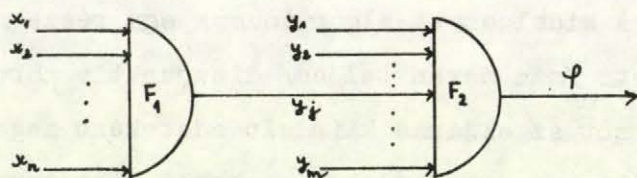
A műszaki gyakorlatban egy Boole-függvényt realizáló hálózat egy olyan struktúra, amely bizonyos Boole-függvényeket realizáló, tetszőleges /véges számú bemenettel és egyetlen kimenettel rendelkező funkcionális elemekből a következő szabályok szerint épül fel:

- a/ Egy elem bemenete vagy a hálózat bemenete, vagy egy másik elem kimenetével van összekötve.
- b/ A hálózatnak egyetlen olyan funkcionális eleme van, amelynek a kimenete az egész hálózat kimenete, a többi elem kimenete valamely másik elem bemenetével van összekötve.
- c/ Egy bemenet csak egy kimenethez csatlakozhat.
- d/ Egy kimenet több bemenettel is összeköthető.
- e/ A hálózat hurokmentes.

Reprezentálják  $F_1$  ill.  $F_2$  funkcionális elemek az  $f_1(x_1, x_2, \dots, x_n)$  ill.  $f_2(y_1, y_2, \dots, y_m)$  Boole-függvényeket. Ha az  $F_1$  funkcionális elem kimenetét az  $F_2$  funkcionális elemnek az  $f_2$  Boole-függvény  $y_j$  változóját reprezentáló

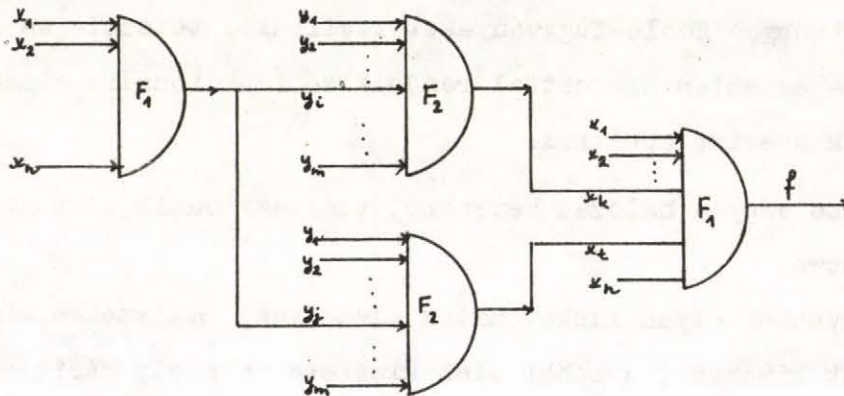


bemenetére kötjük, akkor a két elemből így előálló kombinatorikus hálózat /2. ábra/ a  $\varphi = f_2(y_1, y_2, \dots, y_{j-1}, f_1(x_1, x_2, \dots, x_n), y_{j+1}, \dots, y_m)$  függvényt reprezentálja.



2. ábra

A 3. ábrán látható  $F_1$  és  $F_2$  funkcionális elemekből álló hálózat, - amely az a-e szabályok mindegyikének eleget tesz - az  $f = f_1(x_1, x_2, \dots, x_{k-1}, f_2(y_1, y_2, \dots, y_{i-1}, f_1(x_1, x_2, \dots, x_n), y_{i+1}, \dots, y_m), x_{k+1}, \dots, x_{t-1}, f_2(y_1, y_2, \dots, y_{j-1}, f_1(x_1, x_2, \dots, x_n), y_{j+1}, \dots, y_m), x_{t+1}, \dots, x_n)$  függvényt reprezentálja. Vegyük észre, hogy ha  $f_1$  az  $n$  változós  $\vee$ ,  $f_2$  az  $m$  változós  $\wedge$  függvény, az  $f$  egy zárójeles formulával írható le.



3. ábra

Ily módon egy tetszőleges szerinti hálózat a benne szereplő funkcionális elemek által reprezentált Boole-függvényeknek a hálózat megszabta szuperpozícióját jelenti.

Az építőelemként használt funkcionális elemkészletre általában csak az a kikötés, hogy az elemkészlet által reprezentált Boole-függvények véges



számu felhasználásával tetszőleges  $n$  változós Boole-függvény előállítható legyen szuperpozíció segítségével. Azt mondjuk, hogy egy ilyen függvényhalmaz funkcionálisan teljes rendszert alkot vagy, hogy a Boole-függvények egy bázisát /bázishalmazát/ alkotja. Ebből világos, hogy a bázis fogalmát nem a szokásos módon használjuk, ugyanis ennek a bázishalmaznak nem kell minimálisnak lennie, hanem az további függvények hozzávételével bővíthető, sőt ez a bővítés a realizálás szempontjából célszerű is lehet /40. definíció/.

Korábban a bázisfüggvények problémája nem vetődött így fel. Ugyanis az itéletkalkulusban részletesen ki voltak dolgozva az  $\{\wedge, \vee, \neg\}$  bázishalmazra, amely egyébként szintén nem minimális, vonatkozó eredmények és a kombinatorikus áramkörök tervezésében ezeket az eredményeket alkalmazták. Ez a magyarázata annak is, hogy a minimalizálási probléma rendszerint a diszjunktív vagy konjunktív normálformákban adott Boole-függvénynek ugyanilyen normálformában való minimalizálását jelentette. A legtöbb minimalizáló eljárás a Boole-függvények diszjunktív normálformájában szereplő változó előfordulások számát minimalizálja. E minimalizáló algoritmusok egy része kifejezetten kézi alkalmazásra, kis változószámra /max. 6 változó/ való [67 34]. Alapkövetelmény az abszolút minimum meghatározása. E minimum megtalálása rendkívül időigényes még számítógépen is, mivel szükséges hozzá az összes irredundáns formula megadása [19, 56, 57]. Az  $n$  változós Boole-függvények irredundáns diszjunktív normálformái számára vonatkozóan érvényes a Zsuravljev [71] által adott becslés

$$5^{2^{n-4}} \leq \mathcal{T}(n) < \left( \frac{2^{n+\frac{1}{2}}}{\sqrt{\pi n}} \right) 2^n$$

ahol  $\mathcal{T}(n) = \max_{f \in N} \mathcal{T}_f$ ,  $N$  az  $n$  változós Boole-függvények halmaza és  $\mathcal{T}_f$  az  $f$  függvény összes irredundáns DNF-inak száma. Ez a becslés, még az alsó határt tekintve is mutatja, hogy  $n > 10$  esetén  $\mathcal{T}(n)$  rendkívül nagy szám. Ezért is, de nem alapvetően ezért, nagy változószám esetén nem lehet cél



az abszolút minimum megadása. Ennek alapvető oka azonban az, hogy a kombinatorikus hálózatban szereplő funkcionális elemekre fennálló technológiai korlátozások sokszor nem hangolhatók össze a minimális diszjunktív normálformát realizáló hálózatbeli követelményekkel. Ezért a tervezők olyan anyag elkészítését igénylik, amelynek alapján viszonylag könnyen össze tudnak állítani egy megfelelő formulát. Erre a célra alkalmas a teljes primimplikáns lista, megjelölve a lényeges primimplikánsokat, valamint egy olyan lefedési táblázat, amely megadja, hogy az egyes /nem lényeges/ primimplikánsok által le nem fedett pontjait fedik. Ezért a dolgozatban megadott algoritmusok, vagy algoritmus általánosítások végcélja ilyen eredmény kialakítása.

Meg kell még jegyezni, hogy az eddigi minimalizáló eljárások közül azok, amelyek elvben akárhány változós függvény minimalizálására alkalmasak, csak valamely irredundáns diszjunktív normálforma felírását tűzik ki célul [36, 66, 69].

Ismertek minimalizáló eljárások az  $\{\wedge, \vee, \neg\}$  bázisban felírt zárójeles formulákra [1, 5, 35, 41], de általában vagy nem eléggé hatékonyak és közel járnak a minden eset kipróbálásához, vagy csak kis változószámra alkalmasak, vagy csak egy mélységű zárójelezést használnak.

A dolgozat három fejezetből áll. Az első fejezetben áttekintjük a disszertáció tárgykörével kapcsolatos elméleti kérdéseket. Bevezetjük a disszertációban használt új fogalmakat. Tisztázunk egy sor elméleti problémát. Ismertetjük az új elméleti eredményeket.

A II. fejezetben az  $\{\wedge, \vee, \neg\}$  bázisban diszjunktív normálformában felírt formulákat minimalizáló eljárásokkal foglalkozunk.

A II.1. pontban rövid áttekintést adunk a diszjunktív normálformában minimalizáló ismert eljárásokról, ahol az eljárások részletezése helyett inkább az elveket és a fejlődést igyekszünk bemutatni.



A II.2. pontban két, a II.3. pontban egy további új algoritmust ismertettünk. Két algoritmus a kipróbálás ill. a consensus módszer /lásd: II.1. pont/ alap gondolatát használja fel. Az első algoritmusban /amely több rész-algoritmusból áll/ a többi kipróbálási elven alapuló algoritmussal szemben nem kell a függvény KDNF-ját megadni, mivel az a függvény egy tetszőleges DNF-ja alapján működik, kihasználva annak konkrét formájából adódó lehetőségeket. Megadunk egy olyan eljárást is, amellyel egyidejűleg elő lehet állítani az  $f$  és  $\bar{f}$  ill.  $\Phi$ -Boole-függvényekre /3. definíció/ az  $f_1$  és  $f_0$  összes primimplikánsát. Kritériumot adunk arra, hogy az összes primimplikáns felírásának biztosítása mellett a próbálgatást bizonyos feltételek teljesülése esetén milyen  $k < n$  rangú konjunkciókkal bezárólag lehet abbahagyni. A másik algoritmus /consensus/, amelyet  $\Phi$ -Boole-függvényekre is kiterjesztünk, a memória takarékoság és a műveletek gyorsítása jegyében készült. A memória takarékoságot az összes primimplikáns felírásának két fázisra bontásával, a gyorsítást pedig a vizsgálandó konjunkciópároknak az eltérés itt bevezetett fogalmán alapuló rendkívül egyszerű előzetes összehasonlításával éri el.

A harmadik algoritmus /II.3./ kidolgozásánál párhuzamba állítottuk a nem teljes automaták állapotszám-minimalizálásának módszereit és a Boole-függvények  $\{\wedge, \vee, \neg\}$  bázisban való minimalizálását. A két probléma közötti analógia és az állapotszám minimalizáló eljárások vizsgálata alapján a primimplikánsok felírásához szükséges információ meghatározására kialakítottuk a  $\Phi$ -Boole-függvényekre, valamint az általános Boole-függvényekre az ún. komplementer módon megadott Boole-gráfot. Ez a megadási mód és a primimplikánshoz tartozó pontok közötti, valamint az  $f$  és  $\bar{f}$  implikánsai közötti összefüggések olyan algoritmusok megadását tették lehetővé, amelyekben viszonylag egyszerű az összes primimplikáns felírása és annak eldöntése, hogy egy elemi konjunkció implikáns-e. Az eljárás érdekessége, hogy a teljesen meghatározott és az általános Boole-függvényeket minimalizáló eljárás alig tér el egymástól.



Az algoritmusok befejezése, amely az összes primimplikáns alapján írja fel a lényeges primimplikáns listát és a lefedő táblázatot, közös. Ebben a részben a Boole-függvények pontjain értelmezett ekvivalencia-reláció alapján a Boole-függvény pontjainak ekvivalencia-osztályokba sorolása egyszerűsíti az eredményt.

A III. fejezetben a  $\{\text{NAND}\}$  ill.  $\{\text{NOR}\}$  bázisbeli nem optimális szintézis problémával foglalkozunk. Az  $\{\wedge, \vee, \neg\}$  bázisbeli szintézisre vonatkozó Akers-féle táblázat módszernek [5] kidolgozzuk egy továbbfejlesztését. Ez egy olyan algoritmus, amely  $\{\wedge, \vee, \neg\}$  bázisban bemenetszám-, hálózatmélység- és terhelési korlátozásokat figyelembe vevő szintézist eredményez /határok összehasonlításának módszere/. Ezt a módszert alkalmassá tesszük  $\{\text{NAND}\}$  ill.  $\{\text{NOR}\}$  bázisban való hasonló feltételek melletti szintézisre. Egy újabb általánosítással pedig alkalmassá tesszük ezt az eljárást általános Boole-függvények szintézisére is úgy, hogy az előzőekben említett feltételek mellett az eljárás a közös részfüggvényeket is figyelembe veszi.

A Függelék tartalmazza a kidolgozott algoritmusok blokkvázlatait.

## I. Fejezet

### ALAPFOGALMAK ÉS DEFINÍCIÓK

Ebben a fejezetben röviden áttekintjük a Boole-függvények elméletének azon fogalmait és eredményeit, amelyeket a későbbiekben felhasználunk. A tárgyalás során következetesen figyelembe vesszük a  $B_n$  tér /a  $\{0,1\}^n$  tér/ és az  $n$  változós Boole-függvények tere közötti, az általunk kidolgozott algoritmusokban kihasznált kapcsolatokat, elsősorban azt a tényt, hogy az  $n$  változós Boole-függvények felfoghatók a  $B_n$  tér részhalmazai karakterisztikus függvényeiként. Ily módon a tárgyaláshoz egy olyan szempontot kapunk, amely nagymértékben egyszerűsíti a leírást. A fenti megfontolás alapján bebizonyítunk egy, az eljárásokban felhasznált tételt /ld. 35. old./.

Hálónak nevezünk egy  $H$  halmazt, ha definiálva van benne két művelet, az un. metszés ( $\wedge$ ) és az egyesítés ( $\vee$ ), amelyek  $H$  bármely két  $a, b$  elemének megfeleltetik a  $H$  egy-egy elemét, az  $a$  és  $b$  metszetét  $a \wedge b$  ill. egyesítését  $a \vee b$  [60] úgy, hogy teljesülnek az un. hálóaxiómák:

- |   |   |
|---|---|
| 1./ $(a \wedge b) \wedge c = a \wedge (b \wedge c)$ | 2./ $(a \vee b) \vee c = a \vee (b \vee c)$ |
| 3./ $a \wedge b = b \wedge a$                       | 4./ $a \vee b = b \vee a$                   |
| 5./ $a \wedge (a \vee b) = a$                       | 6./ $a \vee (a \wedge b) = a$               |

ahol  $a, b, c$  a háló tetszőleges elemei.

Látható, hogy az 1.-2., 3.-4., 5.-6. axiómákat úgy kapjuk egymásból, hogy a két hálóműveletet egymással felcseréljük, ami azt jelenti, hogy a metszés és az egyesítés műveletei egymás duálisai.

Ha  $K$  egy  $H$  hálóbeli kifejezés, vagyis a  $H$  háló tetszőleges elemein átfutó változókból az  $\wedge$  és  $\vee$  műveletek akárhányszoros alkalmazásával keletkező kifejezés, akkor felcserélve benne az  $\wedge$  és  $\vee$  műveletjeleket a  $K$  kifeje-







ahol  $\bigwedge_{j=1}^n a_j$  és  $\bigvee_{j=1}^n a_j$  az  $a_1, a_2, \dots, a_n$  elemeknek az 1. és 2. axiómák alapján az algebrában szokott módon definiálható metszete ill. egyesítése.

A tétel bizonyítása során belátjuk, hogy az így definiált  $\leq$  reláció reflexív, antiszimmetrikus és tranzitív, tehát rendezési reláció. A tétel második részéből a hálóműveletek és a hálóbéli rendezés között fennálló egyik kapcsolat könnyen leolvasható. Nevezetesen az, hogy ha  $a_1, a_2, \dots, a_m, b_1, b_2, \dots, b_n \in H$  és minden  $j, k$ -ra ( $j=1, 2, \dots, m, k=1, 2, \dots, n$ )  $a_j \leq b_k$ , akkor  $\bigvee_{j=1}^m a_j \leq \bigwedge_{k=1}^n b_k$ .

A 9. összefüggés alapján nyilvánvaló, hogy az  $a \leq b \equiv a \vee b = b$  definíció is ugyanazt a rendezési relációt adja.

Az eddigiekből következik, hogy ha egy  $H$  halmazban értelmezzünk egy  $R$  rendezési relációt és  $H$  bármely kételemű részhalmazának van infimuma és szuprémuma  $R$ -re nézve, akkor az  $a \wedge b = \inf \{a, b\}$ ,  $a \vee b = \sup \{a, b\}$ -vel definiált műveletekre érvényesek a hálóaxiómák [60]. Ezt úgy szokás kifejezni, hogy  $H$  az  $R$  rendezési relációra nézve hálót alkot.

Ha a hálónak van olyan  $0$  és  $1$  eleme, hogy minden  $x \in H$ -ra  $0 \leq x$  és  $x \leq 1$ , akkor  $0$ -t a háló legkisebb,  $1$ -t a legnagyobb elemének nevezzük. Az  $1$  és  $0$  elemekre fennáll, hogy  $0 \wedge 1 = 0$ ,  $0 \vee 1 = 1$ .

Egy véges  $H$  háló diagramjának nevezzük azt az irányított gráfot, amelynek csúcsai a háló elemeinek felelnek meg, élei pedig következőképpen jellemezhetők: az  $a$ -nak megfelelő csúcsból a  $b$ -nek megfelelő csúcsba ( $a, b \in H$ ) akkor és csak akkor vezet él, ha  $a < b$  és nincs olyan  $c \in H$ , hogy  $a < c < b$  lenne.

Általában, ha  $a \leq b$ , az  $a$ -ból  $b$ -be vezet út. Mivel  $\inf \{a, b\} = a \wedge b$ ,  $\sup \{a, b\} = a \vee b$ , ha  $a \not\leq b$  és  $b \not\leq a$ , akkor az  $a \wedge b$ -ből mind  $a$ -ba, mind  $b$ -be, valamint  $a \vee b$ -be mind  $a$ -ból, mind  $b$ -ből vezet él /4a. ábra/. Ha  $a \leq b$ , akkor  $a$ -t  $b$ -vel általában több élből álló élsorozat köti össze. Az  $\wedge$  művelet eredménye ebben az esetben az út kezdőpontja ( $a$ ),  $\vee$  műve-

leté az ut végpontja (b) /4b. ábra/. A háló diagramjában az élek irányítása a rendezési reláció illusztrálására szolgál. Egy háló diagramjáról tehát a hálóműveletek eredményei a rendezési reláció alapján leolvashatók, így ez egyenértékű a két művelettábla megadásával.

Ha egy  $H$  hálónak van mind legkisebb, mind legnagyobb eleme, akkor az  $a \in H$  komplementumának nevezzük az olyan  $x \in H$  elemet, amelyre  $a \wedge x = 0$  és  $a \vee x = I$ .

Az olyan hálót, amelyben bármely elemnek létezik komplementere /esetleg több is/ komplementumos hálónak nevezzük. Egyértelműen komplementumos egy háló, ha minden elemének egyetlen komplementuma van. Ha egy egyértelműen komplementumos háló tetszőleges  $a$  elemének komplementumát  $\bar{a}$ -al jelöljük, akkor világos, hogy  $\bar{\bar{a}}$  komplementuma  $a$ , vagyis  $\bar{\bar{a}} = a$ .

A hálók egy fontos osztályát képezik az ún. disztributív hálók, amelyek elemeire teljesülnek még a következő

$$10. \quad a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$$

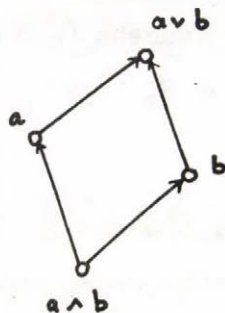
$$11. \quad a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c)$$

disztributivitási egyenlőségek. Általában egy hálóban az

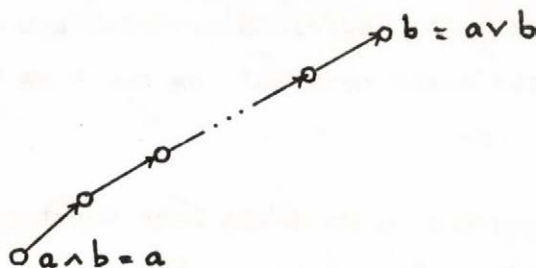
$$a \wedge (b \vee c) \geq (a \wedge b) \vee (a \wedge c)$$

$$a \vee (b \wedge c) \leq (a \vee b) \wedge (a \vee c)$$

ún. disztributivitási egyenlőtlenségek állnak fenn.



a)



b)

4. ábra



Az egyértelműen komplementumos disztributív hálókat Boole-algebráknak nevezzük. Az egyértelműség miatt a komplementum-képzés tulajdonképpen egy egyváltozós művelet, ezért a Boole-algebrák tekinthetők három műveletes algebrai struktúráknak is. Ennek megfelelően a "rész-Boole-algebra" szemben a "részhálóval" az illető Boole-algebra olyan részhálója, amely a komplementum-képzésre is zárt. A Boole-algebra elemeire érvényesek a  $\overline{(a \wedge b)} = \bar{a} \vee \bar{b}$ ,  $\overline{(a \vee b)} = \bar{a} \wedge \bar{b}$  De Morgan-féle azonosságok. A Boole-algebrák tárgyalásakor szokás az  $\wedge$  műveletet konjunkciónak, a  $\vee$  műveletet diszjunkciónak nevezni. A mérnöki gyakorlatban az egyik, általában a  $\vee$  műveletet szokás kitüntetni. Ennek megfelelően a kifejezésekben az  $\wedge$  jelölését általában elhagyják /ugy jelölik, mint az elemi algebrában a szorzatot/. Ezen túlmenően, az egész számok gyűrűjére vonatkozó analógiát tovább vite /ld. pl. [16]/, a Boole-algebra legkisebb ill. legnagyobb elemét szokás nulla ill. egységelemnek nevezni.

Legyen  $H$  egy Boole-algebra. Minthogy  $H$  a definíció értelmében háló is, bármely  $K$  hálóbeli kifejezésre definiálva van annak duálisa,  $D(K)$ . A De Morgan-féle azonosságokból és az  $\bar{\bar{a}}=a$  azonosságból látható, hogy  $D(K)$  /a változók minden értékére/ megegyezik azzal a  $K$  kifejezéssel, amelyet  $K$ -ból úgy kapunk, hogy minden benne szereplő a változót komplementumával,  $\bar{a}$ -al pótoljuk, majd az így kapott kifejezést negáljuk:  $D(K(a,b,...)) = \overline{K(\bar{a},\bar{b},...,)}$ , ahol  $a,b,...$  a  $K$ -ban szereplő változók.

A  $B$  Boole-algebra egyszerű elemének nevezzük az olyan  $x \in B$  elemet, amelyre vagy  $x \wedge y = x$  vagy  $x \wedge y = 0$  minden  $y \in B$ -re. Egyébként  $x$  a háló összetett eleme. Érvényes a következő két tétel [22].

#### Tétel:

Egy véges  $B$  Boole-algebra bármely összetett eleme felbontható a  $B$  bizonyos egyszerű elemeinek diszjunkciójára, a tagok sorrendjétől eltekintve egyértelműen.

#### Tétel:

Egy véges  $B$  Boole-algebra összes egyszerű elemeinek diszjunkciója az al-



gebra egységelemével egyenlő.

Jelölje az "a" elem komplementerét  $\bar{a}$ ; az 1.-6. axiómákból levezethetők az alábbi

$$\begin{array}{ll}
 ax \vee a\bar{x} = a & /1/ \text{ összevonási} \\
 ab \vee a = a & /2/ \text{ elnyelési} \\
 ax \vee ab\bar{x} = ax \vee ab & /3/ \text{ redukálási} \\
 ax \vee b\bar{x} = ax \vee b\bar{x} \vee ab & \\
 abx \vee ac\bar{x} = abx \vee ac\bar{x} \vee abc & /4/ \text{ bővitési}
 \end{array}
 \left. \vphantom{\begin{array}{l} ax \vee a\bar{x} = a \\ ab \vee a = a \\ ax \vee ab\bar{x} = ax \vee ab \\ ax \vee b\bar{x} = ax \vee b\bar{x} \vee ab \\ abx \vee ac\bar{x} = abx \vee ac\bar{x} \vee abc \end{array}} \right\}$$

azonosságok. /4/-ben az ab vagy az abc konjunkciót bővitménynek nevezzük.

A továbbiakban néhány speciális, a disszertációban felhasznált Boole-algebrát mutatunk be. Előljáróban ismertetjük a következő hálóelméleti eredményt.

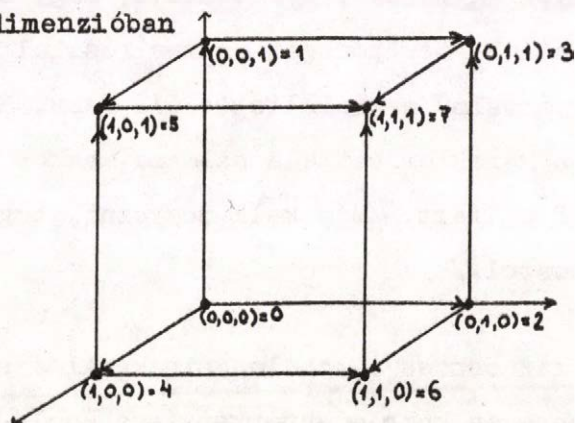
Legyenek  $H_1, H_2, \dots, H_n$  véges hálók. E hálók direkt szorzatán azt a  $H = \prod_{i=1}^n H_i$  halmazt értjük, amelynek elemei a  $(h_1, h_2, \dots, h_n)$  elem n-esek, ahol  $h_i \in H_i$  ( $i=1, 2, \dots, n$ ), és amelyben az  $\wedge, \vee$  művelet és a rendezési reláció komponensenként értendő. Akkor H e műveletekre nézve szintén háló és a benne mint hálóban definiált rendezési reláció ekvivalens a fent definiálttal.

A triviális egyelemű Boole-algebrától eltekintve a legegyszerűbb példa Boole-algebrára a következő. Vesszünk két különböző elemet a-t és b-t és közöttük a rendezési relációt úgy definiáljuk, hogy  $a \leq b$  fennáll. Könnyű belátni, hogy e rendezési relációra nézve  $\{a, b\}$  halmaz Boole-algebrát alkot /lásd pl. [16] /. Ebben a Boole-algebrában  $0=a$ ,  $1=b$ ,  $a \wedge b=a$ ,  $a \vee b=b$ . Ezt a Boole-algebrát kételemű Boole-algebrának nevezzük.

Legyen  $H_i(a_i, b_i)$ , ( $i=0, 1, \dots, n-1$ ), n darab kételemű Boole-algebra. Képezzük ezek direkt szorzatát, a  $H = \prod_{i=1}^n H_i$  hálót, amely Boole-algebra. H elemei az összes lehetséges,  $2^n$  darab,  $a_i, b_j$ -ből álló n elemű sorozat, ahol a k-adik helyen  $a_k$  vagy  $b_k$  áll.



Legyen speciálisan minden  $i$ -re  $a_i=0$ ,  $b_i=1$ , vagyis mindegyik  $H_i=H_0$ , ahol  $H_0$  a  $\{0,1\}$  kételemű Boole-algebra ( $0 \leq 1$ ). Tehát  $n$  számú  $H_0$ -val azonos Boole-algebra direkt szorzata, más szóval az  $n$  dimenziós bit-tér /vagy Boole-tér/ is Boole-algebrát alkot. Mi a továbbiakban erre a  $B_n$  tér elnevezést használjuk. E Boole-algebra hálódiaagramját szokás  $n$  dimenziós Boole-kockának is nevezni. Az  $n$  dimenziós Boole-kocka minden csúcsához  $n$  él tartozik. Pl. 3 dimenzióban



5. ábra

Az így konstruált Boole-algebrák igen egyszerűek, számunkra elsősorban a szemléltetés könnyítését szolgálják.

Nevezzük Boole-változóknak azokat az  $x_1, x_2, \dots$  változókat, amelyek csak két értéket, egyszerűség kedvéért 0-t vagy 1-et vehetnek fel és Boole-konstansoknak a 0 vagy 1 számokat. A továbbiakban változón mindig Boole-változót értünk.

### 1. definíció

Teljesen meghatározott  $n$  változós Boole-függvénynek nevezzük a fenti  $(x_1, x_2, \dots, x_n)$  változókészleten definiált olyan  $f(x_1, x_2, \dots, x_n)$  függvényt, amely a változók minden érték kombinációjára értelmezve van és 0 vagy 1 értéket vesz fel.

A továbbiakban függvényen mindig Boole-függvényt értünk /vagy az 1. definícióban szereplő, vagy későbbi definíciókban megadandó kiterjesztett ér-

telemben/. Világos, hogy az  $x_1, x_2, \dots, x_n$  változók összes lehetséges értékkombinációinak halmaza és a  $B_n$  tér pontjai kölcsönösen egyértelműen leképezhetők egymásra. Tehát egy  $n$  változós Boole-függvény a  $B_n$  téren értelmezett karakterisztikus /vagyis csak a 0 vagy 1 értéket felvevő/ függvényként is felfogható. Ennek alapján megfeleltetve egy  $n$  változós Boole-függvényt és karakterisztikus halmazát /vagyis azt a halmazt, amelynek ő a karakterisztikus függvénye/ egymásnak, nyilvánvaló, hogy az  $n$  változós Boole-függvények halmaza és a  $B_n$  tér pontjai összes részhalmazainak halmaza között kölcsönösen egyértelmű megfeleltetés áll fenn. Ha  $f$  egy  $n$  változós Boole-függvény és  $R$  a karakterisztikus halmaza, akkor azt is fogjuk mondani, hogy  $f$  lefedi az  $R$  halmazt. /Meg kell jegyezni, hogy ez a lefedés fogalom eltér a szokásostól./

Tudjuk, hogy egy véges halmaz összes részhalmazainak halmaza a halmazelméleti metszésre, egyesítésre és komplementerképzésre Boole-algebrát alkot. Ebből következik, hogy az  $n$  változós Boole-függvények halmaza, a karakterisztikus halmazokon értelmezett rendezési relációra nézve egy Boole-algebrát alkot, amelyet a következőkben  $N$ -el jelölünk.

Könnyű belátni, hogy ez a rendezési reláció a következő két hálóműveletet határozza meg:

Vezessük be a  $f=f(x_1, x_2, \dots, x_n)$  és  $\varphi=\varphi(x_1, x_2, \dots, x_n)$

jelölést.

Konjunkció,  $\wedge$ :

$$f \wedge \varphi = \begin{cases} 1 & \text{az } f \text{ és } \varphi \text{ karakterisztikus halmazai metszetének megfelelő értékkombinációkra, tehát akkor és csak akkor,} \\ & \text{ha } f = \varphi = 1. \\ 0 & \text{egyébként.} \end{cases}$$



Diszjunkció,  $\vee$  :

$$f \vee \varphi = \begin{cases} 1 & \text{az } f \text{ és } \varphi \text{ karakterisztikus halmazai egyesítésének} \\ & \text{megfelelő értékkombinációkra, tehát akkor és csak} \\ & \text{akkor, ha } f \text{ és } \varphi \text{ közül legalább az egyik } 1 \text{ értéket} \\ & \text{vesz fel.} \\ 0 & \text{egyébként.} \end{cases}$$

E hálóműveletek természetesen az  $f$  és  $\varphi$  változók /kétváltozós/ Boole-függvényének is tekinthetők.

Értelmezve van továbbá minden  $f$ -re a következő egyváltozós művelet, amelynek értéke  $f$  komplementuma az  $N$  Boole-algebrában:

Negáció,  $\neg$  :

$$\overline{f} = \neg f = \begin{cases} 1 & \text{ha } f=0 \\ 0 & \text{egyébként.} \end{cases}$$

Vagyis az  $f$  függvény negáltjának,  $\overline{f}$ -nek, karakterisztikus halmaza az  $f$  karakterisztikus halmazának komplementere. Felsorolunk ezenkívül néhány fontosabb kétváltozós Boole-függvényt /a változókat ismét  $f$ -fel és  $\varphi$ -vel jelöljük, mert a továbbiakban rendszerint Boole-függvényekre fogjuk a műveleteket alkalmazni/:

Implikáció,  $\rightarrow$  :

$$f \rightarrow \varphi = 1 \text{ akkor és csak akkor, ha } f \wedge \overline{\varphi} = 0$$

Kizárólagos vagy /antivalencia/,  $\oplus$  :

$$f \oplus \varphi = 1 \text{ akkor és csak akkor, ha } f=1 \text{ vagy } \varphi=1, \text{ de } f \wedge \varphi = 0.$$

Ekvivalencia,  $\equiv$  :

$$f \equiv \varphi = 1 \text{ akkor és csak akkor, ha } f=\varphi.$$

NAND,  $/$  :

$$f / \varphi = 1 \text{ akkor és csak akkor, ha } \overline{f \wedge \varphi} = 1.$$

NOR,  $\circ$  :

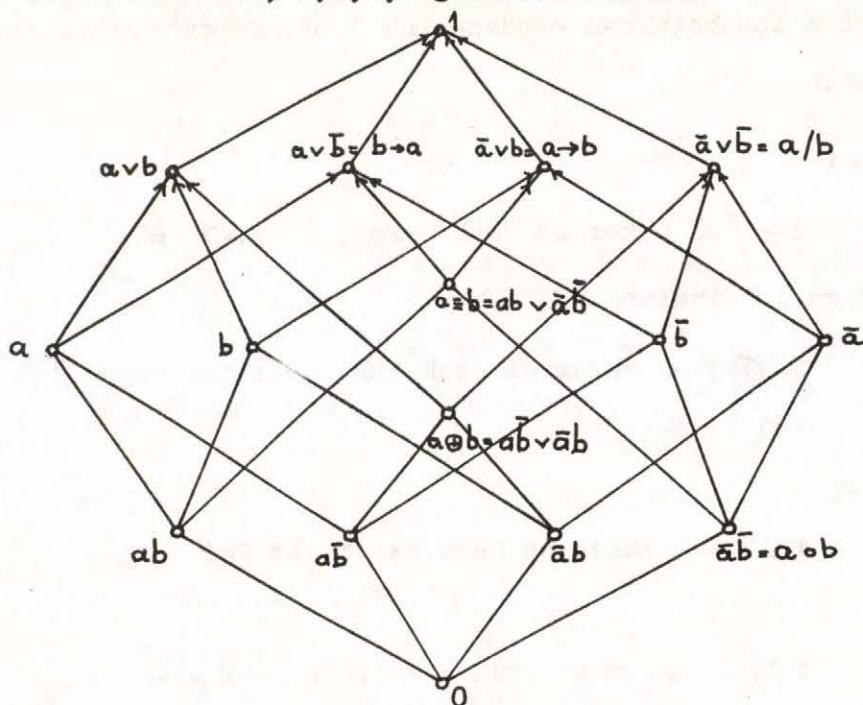
$$f \circ \varphi = 1 \text{ akkor és csak akkor, ha } \overline{f \vee \varphi} = 1.$$

Legyen  $x$  Boole-változó,  $\alpha$  Boole-konstans. Vezessük be az

$$x^\alpha = \begin{cases} \bar{x}, & \text{ha } \alpha=0 \\ x, & \text{ha } \alpha=1 \end{cases}$$

jelölést.

Vegyük az  $x_i^{\alpha_i}$  elemek  $\{x_i^{\alpha_i}\}$  halmazát, ahol  $i=1,2,\dots,n$ -re,  $x_i$  Boole-változó. Szokás  $\{x_i^{\alpha_i}\}$  elemeit az  $N$  Boole-algebra generátorelemeinek is nevezni, mivel alkalmazva rájuk az  $\wedge$  és  $\vee$  műveleteket az  $N$  Boole-algebra minden eleme előállítható. A jelölés figyelembevételével ezt a tényt úgy is megfogalmazhatjuk, hogy az  $N$  Boole-algebra minden eleme /bármely  $n$  változós Boole-függvény/ az  $x_i$  ( $i=1,2,\dots,n$ ) Boole-változókból az  $\{\wedge, \vee, \neg\}$  függvényhalmaz elemeinek szuperpozíciójával - véges sokszor felhasználva ezeket - állítható elő. Ebből következik, hogy az  $\{\wedge, \vee, \neg\}$  függvényhalmaz vehető a  $N$  Boole-algebra bázishalmazaként /ld. 40. definíció/. Illusztrációként bemutatjuk a kétváltozós Boole-függvények Boole-algebrájának hálódíagramját, ahol  $a$  és  $b$  Boole-változók,  $a, b, \bar{a}, \bar{b}$  generátorelemek.



6. ábra



A 14. oldalon kimondott tétel folytán a  $N$ -ben érvényes rendezés értelmében  $x_1^{\alpha_i} \wedge x_j^{\alpha_j} \leq x_1^{\alpha_i}$  és  $\leq x_j^{\alpha_j}$ ,  $x_1^{\alpha_i} \wedge x_1^{1-\alpha_i} = 0$  és  $x_1^{\alpha_i} \vee x_1^{1-\alpha_i} = 1$ . Képezzük az  $x(\alpha_1, \alpha_2, \dots, \alpha_n) = x_1^{\alpha_1} \wedge x_2^{\alpha_2} \wedge \dots \wedge x_n^{\alpha_n}$  konjunkciót, amelyben az  $x_1, x_2, \dots, x_n$  Boole-változók mindegyike pontosan egyszer szerepel. Ha vesszük  $x$ -nek és  $N$  egy tetszőleges elemének konjunkcióját, akkor mint könnyen látható, az eredmény vagy  $x$  vagy  $0$  lesz, tehát  $x$  a  $N$  egyszerű eleme. Az  $N$  összes egyszerű elemét megkapjuk  $x_1^{\alpha_1} \wedge x_2^{\alpha_2} \wedge \dots \wedge x_n^{\alpha_n}$  alakban, kitevőként használva a  $2^n$  különböző  $(\alpha_1, \alpha_2, \dots, \alpha_n)$  érték kombinációt. Az  $N$  egyszerű elemeit szokás mintermeknek is nevezni. Mint az ismert, egy Boole-algebra minden  $0$ -tól különböző eleme egyértelműen előállítható egyszerű elemeinek diszjunkciójaként. Így  $N$  összes nem  $0$  eleme is egyértelműen előállítható mintermek diszjunkciójaként.

Tekintsük a  $N$  egyszerű elemeit. Mivel  $x_1$  értéke  $0$  vagy  $1$  lehet,  $x$  akkor és csak akkor veszi fel az  $1$  értéket, ha  $x_1 = \alpha_1$  minden  $i$ -re. Ennek megfelelően az egyszerű elemek és a  $B_n$  tér pontjai között az  $x_1^{\alpha_1} \wedge x_2^{\alpha_2} \wedge \dots \wedge x_n^{\alpha_n} \leftrightarrow (\alpha_1, \alpha_2, \dots, \alpha_n)$  természetes megfeleltetést fogadjuk el.

## 2. definíció

Legyen  $P = (\alpha_1, \alpha_2, \dots, \alpha_n)$  Boole-konstansokból álló sorozat a  $B_n$  tér egy pontja, amelynek, egyelemű halmazként tekintve, karakterisztikus függvénye az  $x(\alpha_1, \alpha_2, \dots, \alpha_n) = x_1^{\alpha_1} \wedge x_2^{\alpha_2} \wedge \dots \wedge x_n^{\alpha_n}$  konjunkció. E konjunkció tehát lefedi az  $(\alpha_1, \alpha_2, \dots, \alpha_n)$  pontot.

## 3. definíció

Legyen  $P = (\alpha_1, \alpha_2, \dots, \alpha_n)$  a  $B_n$  tér egy pontja,  $f$  pedig tetszőleges  $n$  változós Boole-függvény. Röviden  $f(P)$ -vel jelöljük az  $f(\alpha_1, \alpha_2, \dots, \alpha_n)$  értéket. Az  $f$  függvény 1-pontjának ill. 0-pontjának nevezzük a  $P$  pontot, aszerint, hogy  $f(P) = 1$  ill.  $f(P) = 0$ . A függvény 1 pontjait szokás egyszerűen a függvény pontjainak is nevezni.

A 2. és 3. definícióból következik, hogy ha  $D$  az  $f(x_1, x_2, \dots, x_n)$  Boole-



függvény 1-pontjainak halmaza,  $f(P)=1$  akkor és csak akkor, ha  $P \in D$ .

#### 4. definíció

Elemi konjunkciónak nevezzük a  $\sigma = x_{1_1}^{\alpha_{i_1}} x_{1_2}^{\alpha_{i_2}} \dots x_{1_k}^{\alpha_{i_k}}$  konjunkciót, ha a benne szereplő  $x_{1_j}$  változók mind különbözőek. Egy elemi konjunkció rangja a benne szereplő változók számával egyenlő. Jele  $r(\sigma)$ .

Az  $x_{1_1}^{\alpha_{i_1}} x_{1_2}^{\alpha_{i_2}} \dots x_{1_k}^{\alpha_{i_k}}$  elemi konjunkció 1-pontjai - más szóval az általa lefedett pontok - a  $B_n$  tér azon  $\alpha_1, \alpha_2, \dots, \alpha_n$  pontjai, amelyeknek  $i_1, i_2, \dots, i_k$  koordinátái rögzítettek. A többi  $n-k$  koordináta tetszőlegesen veheti fel a 0 és 1 értéket. Ebből következik, hogy egy  $k$ -rangu elemi konjunkció a  $B_n$  tér egy  $2^{n-k}$  pontból álló olyan részhalmazát fedi le, amely egy  $n-k$  dimenziós Boole-kockát alkot.

#### 5. definíció

A  $B_n$  tér egy  $n-k$  dimenziós kockáját ( $k < n$ ) a következőkben a  $B_n$  tér  $k$ -rangu vagy  $n-k$  dimenziós intervallumának is nevezzük és pontjai fix koordinátáinak ( $k < n$  esetén kihagyásos) sorozatával,  $\alpha'_1, \alpha'_2, \dots, \alpha'_n$ -vel jelöljük, ahol  $j=1, 2, \dots, n$  esetén  $\alpha'_j = \alpha_{1_s}$ , ha  $j=i_s$  és  $\alpha'_j = -/a-jel/$ , ha  $j$  nem fordul elő  $i_1, i_2, \dots, i_k$  között. Ezen intervallumok karakterisztikus függvénye valamely  $x_{1_1}^{\alpha_{i_1}} x_{1_2}^{\alpha_{i_2}} \dots x_{1_k}^{\alpha_{i_k}}$   $k$ -rangu konjunkció.

Az elemi konjunkciók és a  $B_n$  tér intervallumai között a mintermek és a  $B_n$  tér pontjai közti megfeleltetés szellemében az  $(\alpha_1, \alpha_2, \dots, \alpha_n) \longleftrightarrow x_{1_1}^{\alpha_{i_1}} x_{1_2}^{\alpha_{i_2}} \dots x_{1_k}^{\alpha_{i_k}}$  kölcsönösen egyértelmű megfeleltetés áll fenn.

Egy  $\sigma = x_{1_1}^{\alpha_{i_1}} x_{1_2}^{\alpha_{i_2}} \dots x_{1_k}^{\alpha_{i_k}}$  elemi konjunkcióról azt mondjuk, hogy elnyel egy  $\tau = x_{1_1}^{\alpha'_{i_1}} x_{1_2}^{\alpha'_{i_2}} \dots x_{1_n}^{\alpha'_{i_n}}$  mintermet, ha minden olyan  $\alpha'_j$ -re, amelyre  $j=i_s$  ( $s=1, 2, \dots, k$ ),  $\alpha'_j = \alpha_{1_s}$ . A /2/ elnyelési szabályból világos, hogy ez esetben  $\tau \vee \sigma = \sigma$ .

Példa: legyen  $n=5$ ,  $i_1=2$ ,  $i_2=3$ ,  $i_3=5$ ,  $\alpha_{i_1}=1$ ,  $\alpha_{i_2}=0$ ,  $\alpha_{i_3}=1$ .



A megfelelő elemi konjunkció  $x_2\bar{x}_3x_5$  és fennáll a következő megfeleltetés:  
 $(-10-1) \leftrightarrow x_2\bar{x}_3x_5$ . Az  $x_2\bar{x}_3x_5$  konjunkció lefedí a

0	1	0	0	1
0	1	0	1	1
1	1	0	0	1
1	1	0	1	1

pontokból álló halmazt.

A 2. definícióból következik, hogy egy tetszőleges Boole-függvény, amely mint említettük, egyértelműen előállítható bizonyos egyszerű elemek diszjunkciójaként, az ezen előállításban szereplő egyszerű elemek által lefedett  $(\alpha_1, \alpha_2, \dots, \alpha_n)$  pontok halmazának karakterisztikus függvénye. Ennek alapján érvényes az

$$f(x_1, x_2, \dots, x_n) = \bigvee x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n} f(\alpha_1, \alpha_2, \dots, \alpha_n) \quad /5/$$

formula, ahol a diszjunkcióban  $(\alpha_1, \alpha_2, \dots, \alpha_n)$  átfut  $B_n$  pontjain.

## 6. definíció

Az /5/ jobb oldalán álló kifejezést, amely mindazon  $x_1^{\alpha_1} x_2^{\alpha_2} \dots x_n^{\alpha_n}$  mintermek diszjunkciójának is tekinthető, amelyekre  $f(x_1, x_2, \dots, x_n) = 1$ , az  $f$  függvény kitüntetett diszjunktív normálformájának /KDNF/ nevezzük. Tehát egy függvény KDNF-jának megadása lényegében a  $B_n$  tér azon pontjainak felsorolását jelenti, amelyekben a függvény az 1 értéket veszi fel.

Az /1/, /2/ azonos átalakításokat alkalmazva egy függvény KDNF-jára, a függvényt elemi konjunkciók diszjunkciójából álló formában kaphatjuk meg.

## 7. definíció

Egy függvény diszjunktív normálformája /DNF/, olyan elemi konjunkciók diszjunkciója, amelyek csak a függvény pontjait fedik le és a függvény minden pontja legalább egy konjunkciónak pontja. Mivel minden Boole-függ-

vény egy  $R \subset B_n$  ponthalmaz karakterisztikus függvénye, egy függvény DNF-jának megadása a  $R$  pontjaiból alkotott intervallumok felsorolása, ahol  $R$  minden pontja legalább egy intervallumba beletartozik.

E definícióból is világos, hogy egy függvény DNF-ba való írása általában nem egyértelmű, bár KDNF csak egy van.

Nyilvánvaló, hogy ha  $f = \bigvee_{i=1}^l \sigma_i$  az  $f$  függvény diszjunktív normálformája, akkor bármely  $i=1,2,\dots,l$ -re  $\sigma_i \rightarrow f$ , hiszen, ha  $\sigma_i=1$ , akkor  $f=1$  és így  $\sigma_i \wedge \bar{f}=0$  mindig teljesül.

### 8. definíció

Egy függvény valamely diszjunktív normálformájában szereplő elemi konjunkciókat a függvény implikánsainak nevezzük.

### 9. definíció

Egy függvény azon implikánsait, amelyek minimális számú változót tartalmaznak abban az értelemben, hogy a belőlük egy /vagy több/ változó elhagyásával keletkező elemi konjunkciók egyike sem implikánsa a függvénynek, e függvény primimplikánsainak nevezzük.

### 10. definíció

A  $B_n$  tér egy  $R$  részhalmazának pontjaiból alkotható minimális rangú intervallumokat, vagyis azokat, amelyek nem részhalmazai  $R$  pontjaiból álló alacsonyabb rangú intervallumnak, a  $R$  primintervallumainak nevezzük. Könnyen látható, hogy az ezeknek megfelelő konjunkciók éppen a  $R$  karakterisztikus függvényének primimplikánsai.

### 11. definíció

A  $B_n$  tér egy  $R$  részhalmazának azon primintervallumai, amelyeknek van olyan pontja, amelyet csak az illető primintervallum fed le,  $R$  lényeges



primintervallumainak nevezzük. A lényeges primintervallumok megfelelői a lényeges primimplikánsok.

#### 12. definíció

Egy függvény összes primimplikánsának diszjunktíóját a függvény redukált diszjunktív normálformájának /RDNF/ nevezzük.

A 12. definícióból világos, hogy minden függvénynek egyetlen RDNF-ja van.

#### 13. definíció

Legyen  $P$  a függvény primimplikánsainak olyan részhalmaza, hogy

$f(x_1, x_2, \dots, x_n) = \bigvee_{\sigma_i \in P} \sigma_i$ , de bármely  $\sigma_i$ -t elhagyva  $P$ -ből, a maradék primimplikánsok diszjunktíója már nem adja a függvényt; a  $\bigvee_{\sigma_i \in P} \sigma_i$  diszjunktíót a függvény irredundáns diszjunktív normálformájának /IDNF/ nevezzük. Ismeretes, hogy egy függvénynek általában több IDNF-ja van [71].

#### 14. definíció

Az olyan IDNF-át, amelyben minimális számú változóelőfordulás szerepel /minden változót annyiszor számolunk, ahányszor előfordul/ minimális diszjunktív normálformának /MDNF/ nevezzük. Egy függvénynek több MDNF-ja is lehet.

Megjegyzés: A különböző diszjunktív normálformákhoz hasonlóan definiálhatók a megfelelő konjunktív normálformák is. Ezek közül most csak a kitüntetett konjunktív normálformát /KKNF/ említjük meg, amelyben az egyes  $/x_1^{\alpha_1} \vee x_2^{\alpha_2} \vee \dots \vee x_n^{\alpha_n}/$  alakú tényezők minden változót pontosan egyszer tartalmaznak. Ezeket a tényezőket szokás maxtermeknek is nevezni.

Tekintettel arra, hogy a disszertációban általánosabb függvényekkel is foglalkozunk, a tárgyalás rövidítése céljából ezeket a függvényeket a következőkben együtt tárgyaljuk a teljesen meghatározott függvényekkel.



### 15. definíció

Nem teljes vagy parciális, vagy  $\Phi$ -Boole-függvénynek nevezzük az olyan  $n$  változós  $f(x_1, x_2, \dots, x_n)$  függvényt, amelynek értéke a változók bizonyos érték kombinációira nem meghatározott. Tehát a függvény értékei 0 vagy 1 vagy  $\Phi$  /nem meghatározott/ lehetnek. Boole-függvényen /röviden: függvényen/ a továbbiakban teljesen meghatározott, vagy parciális Boole-függvényt értünk.

Az eddig bevezetett műveletek közül az  $\wedge$  és  $\vee$  műveleteket kiterjesztjük tetszőleges  $f, \varphi$  Boole-féle függvényekre, mégpedig úgy, hogy  $f \wedge \varphi = 0$ , ha  $f$  és  $\varphi$  közül legalább az egyik meg van határozva és 0 az értéke /függetlenül attól, hogy a másik meg van-e határozva/;  $f \wedge \varphi = 1$ , ha  $f$  is,  $\varphi$  is meg van határozva és mindkettőnek 1 az értéke; a többi esetben /tehát, ha sem  $f$ , sem  $\varphi$  nincs meghatározva, vagy ha  $f$  és  $\varphi$  közül az egyik meg van határozva és 1 az értéke, a másik nincs meghatározva/  $f \wedge \varphi = \Phi$ . Hasonlóan,  $f \vee \varphi = 1$ , ha  $f=1$  vagy ha  $\varphi=1$ , függetlenül attól, hogy a másik tag értéke 0-e, vagy nincs meghatározva, vagy mindkettő 1-e;  $f \vee \varphi = 0$ , ha  $f$  is,  $\varphi$  is meg van határozva és  $f=0, \varphi=0$ ;  $f \vee \varphi = \Phi$ , ha vagy  $f=\Phi, \varphi=\Phi$ , vagy  $f$  és  $\varphi$  egyikének 0 az értéke, a másiké pedig nincs meghatározva.

### 16. definíció

$f_1(x_1, x_2, \dots, x_n)$  és  $f_2(x_1, x_2, \dots, x_n)$  teljesen meghatározott függvények ekvivalensek:  $(f_1(x_1, x_2, \dots, x_n) \equiv f_2(x_1, x_2, \dots, x_n))$ , ha az  $n$  változó minden egyes érték kombinációjára ugyanazt az értéket veszik fel.

### 17. definíció

Egy  $\Phi$ -Boole-függvénnyel kompatibilis minden olyan teljesen meghatározott Boole-függvény, melynek értéke a  $\Phi$ -Boole-függvény meghatározottsági helyein megegyezik annak értékével, a többi helyen pedig tetszőlegesen veheti fel a 0 vagy 1 értéket. Nyilvánvaló, hogy ha egy  $\Phi$ -Boole-függvény  $k$  helyen nincs meghatározva, a vele kompatibilis teljesen meghatározott



Boole-függvények száma  $2^k$ .

Legyen  $g(x_1, x_2, \dots, x_n)$  az a teljesen meghatározott Boole-függvény, amely 1 értéket vesz fel ott, ahol  $f(x_1, x_2, \dots, x_n) = 0$ , vagyis az  $f$  meghatározatlansági tartományán, és 0 értéket egyébként.

#### 18. definíció

Az  $f(x_1, x_2, \dots, x_n)$  függvény alsó határának nevezzük az  $\underline{f} = f \wedge \bar{g}$  teljesen meghatározott Boole-függvényt, amely ott veszi fel az 1 értéket, ahol  $f$ , és a 0-t máshol [37].

#### 19. definíció

Az  $f(x_1, x_2, \dots, x_n)$  felső határának nevezzük az  $\hat{f} = f \vee g$  teljesen meghatározott Boole-függvényt, amely ott veszi fel a 0 értéket, ahol  $f$ , és az 1-et minden más helyen [37].

A  $g$ ,  $\underline{f}$ ,  $\hat{f}$  függvényeket teljesen meghatározott Boole-függvényekre is lehet értelmezni. Teljesen meghatározott Boole-függvények esetén  $g \equiv 0$  miatt  $f = \underline{f} = \hat{f}$ . A 18. definíció alapján, valamint a  $g$  függvény definíciója miatt bármely  $f$  Boole-függvénnyel kompatibilis  $h$  függvény felírható  $h = f \vee \gamma g$  alakban, ahol  $\gamma$  teljesen meghatározott Boole-függvény.

#### 20. definíció

$n$  változós általános Boole-függvénynek nevezzük  $M$  db  $n$  változós /teljesen meghatározott vagy  $\emptyset$ / Boole-függvény un. komponens függvény egy  $F = (f_1, f_2, \dots, f_M)$  sorozatát; általános Boole-változónak ill. általános Boole-konstansnak pedig Boole-változók ill. Boole-konstansok egy  $(x_1, x_2, \dots, x_m)$  ill.  $(\alpha_1, \alpha_2, \dots, \alpha_k)$  sorozatát nevezzük.

#### 21. definíció

Egy  $\Phi$ -Boole-függvény implikánsa olyan  $\sigma$  elemi konjunkció, amely implikálja a függvényt abban az értelemben, hogy minden olyan pontban, ahol  $\sigma = 1$ ,



vagy  $f=1$ , vagy  $f=\Phi$ , és elnyel legalább egy, a függvény 1 pontját lefedő mintermet.

## 22. definíció

Egy  $(f_1, f_2, \dots, f_M)$  általános Boole-függvény implikánsai azok az elemi konjunkciók, amelyek az  $f_1, f_2, \dots, f_M$  függvények közül legalább egynek implikánsai. E függvények egy  $k$  elemű ( $1 \leq k \leq M$ ) részhalmazához tartozó minden egyes  $f_{i_1}, f_{i_2}, \dots, f_{i_k}$  függvényt implikáló elemi konjunkció az illető  $k$  darab függvény együttes implikánsa. Az ilyen együttes implikánsokat bármely is az  $R$  részhalmaz, az  $(f_1, f_2, \dots, f_M)$  általános Boole-függvény együttes implikánsainak nevezzük.

## 23. definíció

Egy  $\Phi$ -Boole-függvény primimplikánsai ill. egy általános Boole-függvény együttes primimplikánsai azok a minimális rangú elemi konjunkciók /ugyanabban az értelemben, mint a 9. definícióban/, amelyek a fenti függvényeknek implikánsai, ill. együttes implikánsai /21., 22. definíció/.

## 24. definíció

Egy általános Boole-függvény lényeges primimplikánsa a függvény egy olyan együttes primimplikánsa, amely a hozzátartozó függvények közül legalább egynek legalább egy pontját egymaga fedi le.

Az  $n$  változós Boole-függvények  $N$  Boole-algebrájának minden  $f$  eleme /pl.  $f$  KDNF-ja útján/ előállítható  $x_1, x_2, \dots, x_n, \bar{x}_1, \bar{x}_2, \dots, \bar{x}_n$ -ből az  $\wedge$  és  $\vee$  műveletek alkalmazásával keletkező  $K = K(x_1, x_2, \dots, x_n, \bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$  kifejezés alakjában, úgy, hogy bármely  $x_1, x_2, \dots, x_n$  pontban  $f(x_1, x_2, \dots, x_n) = K(x_1, x_2, \dots, x_n, \bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$ .  $K$  úgy tekinthető mint a  $H_0 = \{0, 1\}$  háló elemein átfutó  $x_1, x_2, \dots, x_n, \bar{x}_1, \bar{x}_2, \dots, \bar{x}_n$  változókból az  $\wedge$  és  $\vee$  műveletek alkalmazásával keletkező hálóbeli kifejezés és így  $K$  duálisa,  $K^* = D(K)$  definiálva van, és pedig



## 25. definíció

$K^*(x_1, x_2, \dots, x_n, \bar{x}_1, \bar{x}_2, \dots, \bar{x}_n) = K(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n, x_1, x_2, \dots, x_n) =$   
 $= f(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$ . Ez egyúttal azt is mutatja, hogy  $K^*(x_1, x_2, \dots, x_n,$   
 $\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$  értéke független attól, hogy az  $f(x_1, x_2, \dots, x_n)$ -et előállító  
 $K(x_1, x_2, \dots, x_n, \bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$  hálóbeli kifejezést hogyan választottuk. Ez  
a körülmény módot ad  $N$  elemei duálisának definíciójára a következőképpen:  
 $f(x_1, x_2, \dots, x_n)$  duálisa  $f^*(x_1, x_2, \dots, x_n) = f(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)$ .

E definíció alapján, amelyet  $\Phi$ -Boole-függvényekre is kiterjesztünk, világos, hogy  $f^*(x_1, x_2, \dots, x_n)$  duálisa ismét  $f(x_1, x_2, \dots, x_n)$ . Továbbá  
 $(f(x_1, x_2, \dots, x_n) \wedge g(x_1, x_2, \dots, x_n))^* = f^*(x_1, x_2, \dots, x_n) \vee g^*(x_1, x_2, \dots, x_n)$  és  
 $(f(x_1, x_2, \dots, x_n) \vee g(x_1, x_2, \dots, x_n))^* = f^*(x_1, x_2, \dots, x_n) \wedge g^*(x_1, x_2, \dots, x_n)$ .  
Ennélfogva, ha  $f(x_1, x_2, \dots, x_n) \leq g(x_1, x_2, \dots, x_n)$ -t tetszőleges Boole-függvényre úgy definiáljuk, hogy  $f(x_1, x_2, \dots, x_n) \wedge g(x_1, x_2, \dots, x_n) =$   
 $= f(x_1, x_2, \dots, x_n)$  /ami azzal ekvivalens, hogy  $f(x_1, x_2, \dots, x_n) \vee$   
 $\vee g(x_1, x_2, \dots, x_n) = g(x_1, x_2, \dots, x_n)$  akkor  $f(x_1, x_2, \dots, x_n) \leq g(x_1, x_2, \dots, x_n)$   
esetén  $f^*(x_1, x_2, \dots, x_n) \vee g^*(x_1, x_2, \dots, x_n) = f^*(x_1, x_2, \dots, x_n)$ , vagyis  
 $g^*(x_1, x_2, \dots, x_n) \leq f^*(x_1, x_2, \dots, x_n)$ . E szerint a dualitás olyan leképezés,  
amely a rendezési relációt megfordítja és kétszeri alkalmazása az identikus  
leképezés. A dualitás e tulajdonságaiból következik, hogy az  $N$  Boole-algebra  
ra legnagyobb és legkisebb eleme egymás duálisa. Speciálisan az  $0$  ill.  $1$   
Boole-konstansok, mint az  $n=1$  esetnek megfelelő kételemű Boole-algebra leg-  
kisebb ill. legnagyobb elemei, egymás duálisai és egyben egymás negáltjai  
is  $0^* = 1 = \bar{0}$  ill.  $1^* = 0 = \bar{1}$ . Egy  $\alpha_1, \alpha_2, \dots, \alpha_n = A$  általános Boole-konstans  
duálisa ennek megfelelően az egyes komponensek duálisaiból álló  
 $A^* = (\alpha_1^*, \alpha_2^*, \dots, \alpha_n^*) = (\bar{\alpha}_1, \bar{\alpha}_2, \dots, \bar{\alpha}_n) = \bar{A}$  általános Boole-konstans, a-  
mely a  $B_n$  térben az  $(\alpha_1, \alpha_2, \dots, \alpha_n)$  ponttól legtávolabb lévő pont, azaz  
az  $(\alpha_1, \alpha_2, \dots, \alpha_n)$  pontnak a  $B_n$  tér /a  $B_n$ -hez nem tartozó/ középpontjára az  
 $(\frac{1}{2}, \frac{1}{2}, \dots, \frac{1}{2})$  pontra vonatkozó tükörképe.



## 26. definíció

Önduálisnak nevezzük az  $f$  függvényt, ha  $f^* = f$ .

A 25. definícióbeli  $f^*(x_1, x_2, \dots, x_n) = \overline{f(\bar{x}_1, \bar{x}_2, \dots, \bar{x}_n)}$  egyenlőség és az általános Boole-konstans duálisáról elmondottak alapján teljesen meghatározott függvények duálisát a következő szemléletes módon is meghatározhatjuk. Legyen  $B_n$  az  $f(x_1, x_2, \dots, x_n)$  értelmezési tartománya,  $E \subset B_n$  az  $f$  függvény karakterisztikus halmaza,  $\neg(E)$  az  $E$  halmaz komplementere,  $E^* \subset B_n$  az  $E$  elemei duálisainak halmaza, akkor  $\neg(E^*)$  az  $f^*(x_1, x_2, \dots, x_n)$  karakterisztikus halmaza. Szavakban: a  $B_n$  tér pontjai közül elhagyva az  $f(x_1, x_2, \dots, x_n)$  1-pontjainak duálisait, az  $f^*(X)$  karakterisztikus halmazát kapjuk.  $\Phi$ -Boole-függvények esetén a duális függvény karakterisztikus halmaza  $\neg(E^* \cup \emptyset^*)$ , ahol  $\emptyset$  a függvény meghatározatlansági pontjainak halmaza.

Az önduális függvények 1-pontjainak száma ennek megfelelően pontosan  $2^{n-1}$ ; és egyetlen 1-pont tükörképe sem 1-pontja az eredeti függvénynek. Mivel a  $B_n$  tér pontjai közül  $2^{2^{n-1}}$ -féleképpen lehet  $2^{n-1}$  számú pontot úgy kiválasztani, hogy a kiválasztott pontok duálisai mind nem kiválasztottak legyenek, az önduális függvények száma  $2^{2^{n-1}}$ .

## 27. definíció

Az  $f(x_1, \dots, x_n)$   $n$  változós függvény /ténylegesen/ függ az  $x_1$  változótól, ha van olyan értékkombinációja az  $x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n$  változóknak, hogy  $f(x_1, \dots, x_{i-1}, 0, x_{i+1}, \dots, x_n) \neq f(x_1, \dots, x_{i-1}, 1, x_{i+1}, \dots, x_n)$ . Az  $f$  függvény csak olyan DNF-ban írható fel, amelyben az  $x_1$  szerepel.

## 28. definíció

Legyen  $f$  az  $x_1$ -től ténylegesen függő  $\Phi$ -Boole-függvény. Az  $x_1$ -től nem függő,  $f$ -fel kompatibilis  $g$   $\Phi$ -Boole-függvényt  $f$   $x_1$  szerinti lezárásának nevezzük [37. I.50].

## 29. definíció

Egy  $f(a, x)$  Boole-függvény monoton az " $a$ " változóban, ha  $f(a, x)$  felírható



olyan DNF-ban, amelyben vagy csak az  $a$ , vagy csak az  $\bar{a}$  szerepel. Ha ebben a DNF-ban csak az  $a$  változó fordul elő,  $a$ -ban növekvőnek, ha csak az  $\bar{a}$  fordul elő,  $a$ -ban csökkenőnek nevezzük  $f(a, x)$ -et.

### 30. definíció

Legyen  $g$  az  $a$ -ban nem növekvő függvény, akkor  $g$ -nek  $a$ -ban növekvő felső burkolója az a  $\varphi_a$  függvény, amelyre  $\varphi_a \geq g$  és minden  $a$ -ban növekvő  $\psi$  függvényre  $\psi \geq g \rightarrow \psi \geq \varphi_a$

A  $g$   $a$ -ban növekvő alsó burkolója a legnagyobb olyan  $\psi_a$  függvény, amelyre  $\psi_a \leq g$  /lásd [37] VI.6,8/.

A továbbiakban a  $B_n$  tér pontjai közötti kapcsolatokat vizsgáljuk.

### 31. definíció

A  $B_n$  tér két  $P_1, P_2$  pontjának  $\overline{P_1 P_2}$  távolságán különböző koordinátáiknak számát értjük /Hamming-féle távolság/.

A  $B_n$  tér intervallumai közötti eltérést a pontok közt értelmezett távolságfogalommal analóg módon definiáljuk

### 32. definíció

Két  $k_1$  és  $k_2$  rangú intervallum eltérésén a mindkét intervallumban rögzített koordináták közül a különbözők számát értjük. Jele  $\varphi(a_1, a_2)$ , ahol  $a_1$  és  $a_2$  a szóban forgó intervallumok vagy a megfelelő konjunkciók. Pl.  $a_1 = x_1 \bar{x}_2 x_3 x_5 x_7$  és  $a_2 = \bar{x}_2 \bar{x}_3 x_4 x_5$  konjunkcióknak megfelelő intervallumokra  $\varphi(a_1, a_2) = 1$ .

### 33. definíció

Az 0 eltérésű intervallumokat hasonlóknak nevezzük.

### 34. definíció

Az 1 eltérésű intervallumokat szomszédosoknak nevezzük.

Megjegyzés: A 32. definícióval bevezetett eltérés nem mérték, mivel, kivé-

ve azt az esetet, amikor a két intervallum ugyanazon koordinátái rögzítettek, a távolságaxiómák közül [6] csak a szimmetria-axiómát elégíti ki. Ezen eltérésfogalom nem elégíti ki sem az identitás-axiómát, azaz két, 0 eltérésű intervallum nem feltétlenül azonos /pl.  $x_1x_2x_3$  és  $x_1x_3x_5x_7$  -nek megfelelő különböző intervallumok eltérése 0/, sem a háromszögaxiómát /pl.  $a_1=\bar{x}_1x_3x_4x_5$ ,  $a_2=x_1x_2$ ,  $a_3=x_1\bar{x}_2\bar{x}_4\bar{x}_5$ , esetén  $\varphi(a_1,a_2)+\varphi(a_2,a_3)<\varphi(a_1,a_3)$ /. Viszont ez az eltérésfogalom igen szemléletes abból a szempontból, hogy megadja az egyes intervallumoknak, mint pontthalmazoknak a távolságát. Az eltérés ugyanis 0, ha a két intervallumnak van legalább egy közös pontja, és általában az eltérés  $l$ , ha a két intervallum legközelebbi pontjainak távolsága  $l$ . Ebben a terminológiában a /4/ bővítési szabály szemléletesen a  $bx$  és  $c\bar{x}$  szomszédos intervallumokban foglalt azonos rangú, 1 eltérésű  $bcx$  és  $bc\bar{x}$  részintervallumok összevonásával kapott  $bc$  intervallum hozzávételét jelenti.

### 35. definíció

Az  $a$  és  $b$  intervallumok összeköthetők, ha van olyan véges  $a_i$  ( $i=1,2,\dots,n$ ) intervallumsorozat, hogy

$$a = a_1, b = a_n \text{ és}$$

$$\varphi(a_i, a_{i+1}) = 0 \text{ minden } i\text{-re } (i=1,2,\dots,n-1).$$

### 36. definíció

A  $B_n$  tér  $n$  pontjából álló olyan pontsorozatot, ahol az  $i$ -edik és  $i+1$ -edik ( $i=1,2,\dots,n-1$ ) pontok távolsága 1, utnak nevezzük.

### 37. definíció

A  $B_n$  tér egy részhalmazát összefüggőnek nevezzük, ha tetszőleges két pontjához van legalább egy olyan ut, amelyben mindkét pont benne van.

Megjegyzés: Könnyű belátni, hogy egy  $f$  Boole-függvény azon primimplikánsai, amelyek a  $B_n$  tér egy összefüggő részhalmazára intervallumainak felelnek meg,



összeköthetők /ld. 35., 37. definíciók/. Így, ha az  $f$  által lefedett halmaz nem összefüggő, akkor  $e$  halmaz összes primintervallumainak halmaza szétesik olyan részhalmazokra, hogy az egy részhalmazhoz tartozó intervallumok összeköthetők, különböző részhalmazokhoz tartozó intervallumok nem köthetők össze.

### 38. definíció

Egy  $f$  Boole-függvény diszjunkt tartományai - mint azt a következő tétel is mutatja - a  $B_n$  térbeli olyan közös pont nélküli maximális ponthalmazok, amelyek egyesítése az  $f$  által lefedett  $R$  halmaz, és amelyekből csak egyesítéssel minden primimplikánsnak megfelelő intervallum előállítható. Egy függvény primimplikánsainak ismeretében a függvény diszjunkt tartományai egyszerűen meghatározhatók. Vezessük be a konjunkciókra is a Boole-változókra már bevezetett jelölést: ha  $p$  egy konjunkció és  $\alpha$  Boole-konstans, akkor

$$p^\alpha = \begin{cases} p, & \text{ha } \alpha = 1 \\ \bar{p}, & \text{ha } \alpha = 0 \end{cases}$$

Legyen  $p_1, p_2, \dots, p_m$  az  $f$  összes primimplikánsa. Vegyük a 0, 1-ekből álló  $2^m - 1$  számú összes lehetséges olyan  $(\alpha_1, \alpha_2, \dots, \alpha_m)$  értékkombinációt, amelynek nem minden eleme 0 és képezzük a  $d_j = p_1^{\alpha_1} p_2^{\alpha_2} \dots p_m^{\alpha_m}$  kifejezéseket  $(j = \sum_{k=1}^m \alpha_k 2^{m-k})$ .

### Tétel:

A 0-tól különböző  $d_j$  függvények által lefedett ponthalmazok, ahol  $j=1, 2, \dots, 2^m - 1$ ,  $f$  diszjunkt tartományait alkotják.

### Bizonyítás:

Mivel  $p_i = 1$  azokra a pontokra, amelyeket  $p_i$  lefed,  $\bar{p}_i = 1$  azokra a pontokra, amelyeket  $p_i$  nem fed le, ezért egy adott  $j$ -re  $d_j = 1$  azokra a pontokra, amelyeket a  $d_j$ -ben  $\alpha = 1$  kitevővel /negátatlanul/ szereplő primimplikánsok mindegyike lefed, de más primimplikáns nem fed le. Ha ilyen pontok



nincsenek, akkor minden pontban  $d_j=0$ . Ebből következik, hogy:

a./  $d_j=1$  csak az  $f$  pontjaira lehet. /A  $p_i$ -k csak  $f$  pontjait fedik./

b./  $d_j \cdot d_k=0$ , ha  $j \neq k$ . /Van legalább egy  $p_i$ , amely a  $d_k$ -ban negátatlanul, a  $d_j$ -ben negálva szerepel./.

c./  $f$  tetszőleges primimplikánsa felírható  $p_i = \bigvee_{s=1}^{k_i} d_{j_s}$  alakban.

/Ti. az összes olyan  $d_j$  diszjunkciója, amelyben a  $p_i$  negátatlanul szerepel./

d./  $f = \bigvee_{j=1}^{2^m-1} d_j$ , ugyanis bármelyik  $d_j$ -ben van legalább egy negátatlan  $p_i$ , ami biztosítja, hogy  $d_j=1$  csak  $f$  által lefedett pontokra áll.  $f$  minden pontját lefedi valamelyik  $d_j$ , nevezetesen az, ahol az illető pontot lefedő primimplikánsok negátatlanul, a többiek negálva szerepelnek.

e./ Belátjuk, hogy  $d_j$  pontjainak száma nem növelhető a diszjunkt tartományok tulajdonságainak fenntartása mellett. A konstrukcióból következik, hogy  $d_j$  lefedi az összes olyan pontot, amelyet minden, a  $d_j$ -ben negátatlanul szereplő primimplikáns lefed, és egyetlen, a  $d_j$ -ben negálva szereplő primimplikáns sem fed le. Hozzávéve  $f$  egyetlen pontját egy  $d_j$  által lefedett ponthalmazhoz, biztosan van legalább egy olyan  $p_k$  primimplikáns, amely ezt a pontot fedi és  $d_j$  pontjait nem, vagy ezt a pontot nem fedi, de  $d_j$  pontjait igen. Mindkét esetben lehetetlen a fenti  $p_k$  primimplikáns  $p_k = \bigvee_{s=1}^{l_k} d_{j_s}$  alakú előállítás. Az első esetben a  $p_k$  pontjain kívül szerepelni fognak  $d_j$  pontjai is, a második esetben pedig  $p_k$  pontjain kívül szerepelni fog a  $d_j$  pontjaihoz hozzávett pont. Tehát a  $d_j (j=1,2,\dots,2^m)$  által lefedett ponthalmaz maximális. Ezzel a tételt bebizonyítottuk.

A fentiekben megadott konstruktív bizonyítás lépéseit az egyes algoritmusokban követjük.

A továbbiakban meg fogjuk vizsgálni a  $N$  Boole-algebra elemeinek az  $\wedge, \vee, \neg$  műveletektől különböző műveletek segítségével való előállításának néhány kérdését.



### 39. definíció

Az  $f_p(x_1, x_2, \dots, x_1, \dots, x_n)$  és  $f_q(x_1, x_2, \dots, x_n)$  Boole-függvények szuperpozícióján értjük a  $f_p(x_1, x_2, \dots, f_q(x_1, x_2, \dots, x_n), \dots, x_n)$  függvényt, amelyet az  $f_q$  függvénynek az  $f_p$  függvény valamely  $x_1$  argumentuma helyébe történő behelyettesítése útján kapunk.

### 40. definíció

A  $N$  Boole-algebra bázishalmazának nevezünk egy  $\{f_1, f_2, \dots, f_k\}$  függvényhalmazt akkor, ha a  $N$  Boole-algebra bármely eleme előállítható az  $x_i (i=1, 2, \dots, n)$  Boole-változókból az  $f_1, f_2, \dots, f_k$  függvények segítségével, szuperpozíciók véges számú alkalmazásával. Szokás a bázishalmaz elemeiről azt mondani, hogy azok egy funkcionálisan teljes függvényrendszert alkotnak.

Megjegyzés: A definícióból látható, hogy a bázishalmaznál ill. a funkcionálisan teljes függvényrendszernél a minimalitás nem követelmény.

### 41. definíció

A Boole-függvények egy halmazát a Boole-függvények egy zárt osztályának nevezzük, ha a függvényhalmazból a szuperpozíció nem vezet ki.

Post kimutatta, hogy a következő függvényosztályok mindegyike zárt függvényosztályt alkot.

1. A lineáris függvények osztálya
2. A monoton függvények osztálya
3. A 0 /konstanst/ tartó függvények osztálya
4. Az 1 /konstanst/ tartó függvények osztálya
5. Az önduális függvények osztálya

Az előzőekben már definiáltuk az önduális /26. definíció/ és az egy bizonyos változóban monoton /29. definíció/ függvényeket. A monoton függvények minden változójukban monotonok.



#### 42. definíció

Lineáris függvénynek nevezünk egy  $f(x_1, \dots, x_n)$  Boole-függvényt akkor, ha  $f = a_0 \oplus a_1 x_1 \oplus \dots \oplus a_n x_n$ , ahol  $a_i$  ( $i=1, 2, \dots, n$ ) Boole-konstans.

#### 43. definíció

0 tartónak ill. 1 tartónak nevezünk egy  $f$  Boole-függvényt akkor, ha  $f(0, 0, \dots, 0) = 0$  ill.  $f(1, 1, \dots, 1) = 1$ .

Post vizsgálta a zárt függvényosztályok és a funkcionálisan teljes függvényrendszerek kapcsolatát [55].

#### Post tétele:

Annak szükséges és elégséges feltétele, hogy Boole-függvények egy rendszerre funkcionálisan teljes legyen az, hogy legyen benne legalább egy olyan függvény, amely nem lineáris, legalább egy, amely nem monoton, legalább egy, amely nem 0 tartó, legalább egy, amely nem 1 tartó, legalább egy, amely nem önduális.

Mint azt a bevezetésben kifejtettük, bizonyos Boole-függvényeket reprezentáló funkcionális elemekből felépített hálózat a hálózatban szereplő funkcionális elemek által reprezentált Boole-függvényeknek a hálózat megszabta szuperpozícióját jelenti. Ebből következik, hogy ha egy bázishalmaz minden eleméhez van őt reprezentáló funkcionális elem, e funkcionális elemekből alkotott hálózatokkal bármely Boole-függvény reprezentálható.

Az eddigiekben részletesen foglalkoztunk a Boole-függvényeknek az  $\{\wedge, \vee, \neg\}$  bázishalmaz segítségével történő előállításával /szintézisével/. Azonban az  $\wedge, \vee, \neg$  funkcionális elemek megvalósítása óta a funkcionális elemek gyártási technológiájában rendkívül nagy fejlődés ment végbe. Néhány éve már többféle, e három függvénytől különböző függvényt realizáló funkcionális elemet gyártanak, mint pl. különböző bemenetszámú NOR, NAND,  $\oplus$ ,  $U(a, b, c) = ab \vee \bar{a}c$ , stb. függvényeknek megfelelő elemeket. A jövőben pedig várható egyre bonyolultabb Boole-függvényeket reprezentáló funkcionális elemek



kifejlesztése. Ez a tény egyre inkább előtérbe helyezi a Boole-függvényeknek adott bázisban minimális vagy irredundáns formulákkal történő felírásának kérdését.

Az adott  $\{f_1, f_2, \dots, f_s\}$  bázisbeli szintézis feladata az  $\{\wedge, \vee, \neg\}$  bázis kivételével általánosságban nem megoldott, azonban több megközelítése ismert.

Az egyik megközelítés szerint [54] megadják az  $f$  függvényhez az  $F_1$  és  $\bar{F}_1$  ( $i=1, 2, \dots, 2^n$ ) karakterisztikus függvényeket, ahol

$$F_1 = \begin{cases} 1, & \text{ha } f(\alpha_1, \alpha_2, \dots, \alpha_n) = 1 \text{ az } \sum_{j=1}^n \alpha_j 2^{j-1} = i \text{-re,} \\ 0 & \text{egyébként.} \end{cases}$$

Ezek felhasználásával  $f(x_1, x_2, \dots, x_n) = \bigvee_{\substack{(\alpha_1, \alpha_2, \dots, \alpha_n) \in \{P_1\}}} F_1$  az  $f$  KDNF-ja,  $f(x_1, x_2, \dots, x_n) = \prod_{\substack{(\alpha_1, \alpha_2, \dots, \alpha_n) \in \{P_0\}}} \bar{F}_1$  az  $f$  KKNF-ja, ahol  $\{P_1\}$  ill.  $\{P_0\}$  az  $f$  1- ill. 0-pontjainak halmaza. Az  $\wedge, \vee, F_1$  és  $\bar{F}_1$  függvényeknek az  $\{f_1, f_2, \dots, f_s\}$  bázisban való kifejezésével a formula felírása megoldódik, de a minimális vagy irredundáns formulák felírásának problémája általában megoldatlan marad. Igen elterjedt módszer, hogy felírják a MDNF-át, és ezt az alakot átírják a szóban forgó bázisbeli függvényekkel. Ilyen megközelítéssel több,  $\{\text{NOR}\}$  ill.  $\{\text{NAND}\}$  bázisban való szintézis módszernél találkozunk.

Az adott  $\{f_1, f_2, \dots, f_s\}$  bázisbeli szintézis problémájának egy másik megközelítése lehetséges az  $f$  függvény dekompozíciójának felhasználásával. Természetesen a feladat elvileg az összes formula nem feltétlenül teljes végigvizsgálásával megoldható [43].

#### 44. definíció

Legyenek  $X, Y, Z$  közös elemeket nem tartalmazó, olyan általános Boole-változók, hogy  $X \cup Y \cup Z = \{x_1, x_2, \dots, x_n\}$ . Azt mondjuk, hogy az  $f(X, Y, Z)$  Boole-

függvénynek létezik dekompozíciója a  $g$  és  $h$  Boole-függvényekre nézve, ha  $f(x_1, x_2, \dots, x_n) = g(h(x, Z), Y, Z)$  alakba írható. A dekompozíció diszjunkt, ha  $Z$  üres.

A különböző feltételek melletti dekompozíció feladatával számos kutató foglalkozik [10, 11, 21]. Komolyabb eredmények főleg monoton növekvő függvények dekompozíciójára, továbbá dekompozícióra és diszjunkt dekompozícióra ismertek abban az esetben, amikor a 44. definícióban a  $g$  függvény a diszjunkció [51]. A dekompozíció problémakörére vonatkozó elég részletes összefoglalás található [37, VII]-ben.



## II. Fejezet

### BOOLE-FÜGGVÉNYEK MINIMÁLIS VAGY NEM REDUNDÁNS DISZJUNKTIV

#### NORMALFORMÁINAK ELŐÁLLÍTÁSA

##### II.1. Boole-függvények minimális diszjunktív normálformájának meghatározására szolgáló fontosabb módszerek áttekintése

Az első fejezetben mondottakból következik, hogy egy Boole-függvény MDNF-jának meghatározása a függvény összes DNF-iből azon DNF-inak kiválasztását kívánja, amelyekben a változók száma minimális.

Quine [56] bebizonyította, hogy egy függvény bármely MDNF-ja a függvény bizonyos primimplikánsainak diszjunkciója. Ebből következik, hogy az MDNF kialakításának első lépése lehet a függvény összes primimplikánsának felírása. Ezt követi a lényeges primimplikánsok kiválasztása, amelyek a függvény bármely diszjunktív normálformájában szerepelnek, majd olyan primimplikánsok kiválasztása, amelyeknek és a lényeges primimplikánsoknak a diszjunkciója még implikálja a függvényt és minimális számú változót tartalmaz.

A továbbiakban áttekintjük a DNF-ákra vonatkozó fontosabb minimalizáló módszereket.

##### II.1.1. Veitch-diagram és a Karnaugh-módszer

A Veitch-módszer kifejezetten kézi használatra való, amely 4 változóig könnyen, 6 változóig nem túl nehezen alkalmazható; a változósám további növelése azonban használatát egyre nehezkesebbé teszi. A módszer azon alapul, hogy a  $B_n$  tér pontjait  $n \leq 4$  esetén lehet, a síkban egy  $2^{[n]} \cdot 2^{[n-1]}$ -es mátrixba rendezni úgy, hogy a mátrix /akár sor, akár oszlop sze-



rinti/ szomszédos elemei a  $B_n$  tér szomszédos pontjai /a mátrix azonos sorainak vagy oszlopainak első és utolsó eleme szomszédosnak számít/. Pl. egy ilyen elrendezés  $n=3$  esetén:

		$x_2$		$x_2$	
		<hr/>			
$x_1$		110	111	101	100
$x_1$		010	011	001	000
			<hr/>		
		$x_3$	$x_3$	$x_1$	

7. ábra

$n=4$  esetén:

		$x_2$		$x_2$	
		<hr/>			
$x_1$		1100	1101	1001	1000
		1110	1111	1011	1010
$x_1$		0110	0111	0011	0010
		0100	0101	0001	0000
			<hr/>		
		$x_4$	$x_4$	$x_4$	

8. ábra

A fenti típusú mátrix egyes oszlopaihoz és soraihoz egy vagy két változó, vagy a változó negáltja tartozik. A mátrix szegélyezése alapján mátrix-elemként az illető mátrixelem sorához és oszlopához tartozó változók /vagy negált változók/ konjunkcióját tekintjük, amelyek beírására az egyszerű képzési szabály miatt nincs szükség. A mátrix tulajdonságaiból következik, hogy bármely két szomszédos elemére alkalmazható az /1/-es összevonási azonosság, sőt bármely  $2^k$  elemből álló téglalap elemei összevonhatók egy  $n-k$  rangú elemi konjunkcióvá / $n-k$  rangú intervallummá/. Az összevonás eredménye azon változókból /vagy a változók negáltjából/ álló elemi konjunkció, a-



melyek vagy csak negálva, vagy csak negálatlanul fordulnak elő az érintett sorokban és oszlopokban.

### 1. Példa

A 9. ábrán látható mátrixban \* -gal jelölt elemekhez, azaz a négy pont alkotta kettő rangú intervallumhoz, az  $\bar{x}_1 x_4$  elemi konjunkció tartozik.

	$x_2$				
$x_1$					$x_3$
		*	*		
		*	*		
	$x_4$				

9. ábra

A minimalizáló eljárás lényege a következő:

A táblázatban megjelöljük a függvény pontjait /azaz a függvényt KDNF-ban adjuk meg/.

Megkeressük a lehető legtöbb pontot tartalmazó összes intervallumot. Az ezeknek megfelelő elemi konjunkciók lesznek a primimplikánsok. Azok a konjunkciók, amelyekhez tartozó intervallumnak van olyan pontja, amely más intervallumnak nem pontja, a lényeges primimplikánsok.

### 2. Példa

Írjuk fel a 4. ábrán megadott Veitch-diagrammu 4 változós függvény minimális alakját. Primimplikánsok:  $\bar{x}_3 \bar{x}_4$ ,

	$x_2$				
$x_1$	1		1	1	$x_3$
			1		
			1		
	1	1	1	1	
	$x_4$				

10. ábra

lis alakját. Primimplikánsok:  $\bar{x}_3 \bar{x}_4$ ,  $\bar{x}_1 \bar{x}_3$ ,  $\bar{x}_2 x_4$ ,  $\bar{x}_2 \bar{x}_3$ ; ezek közül lényeges primimplikánsok:  $\bar{x}_3 \bar{x}_4$ ,  $\bar{x}_1 \bar{x}_3$ ,  $\bar{x}_2 x_4$ .

Mint hogy a lényeges primimplikánsok lefedik a függvény pontjait egyetlen MDNF létezik, éspedig:  $\bar{x}_3 \bar{x}_4 \vee \bar{x}_1 \bar{x}_3 \vee \bar{x}_2 x_4$ .

Ezen eljárás a tervezők körében ma is

igen elterjedt, sőt ismeretes általánosítása nemcsak a változószám növelése, hanem a  $\Phi$ -Boole-függvények és általános Boole-függvények egyszerűsítése irányában is /15., 20. definíciók/.

Karnaugh módszere gyakorlatilag azonos a Veitch-módszerrel, mindössze a táblázat szegélyezésében van eltérés. A 10. ábrán lévő Veitch-diagram, mint Karnaugh-tábla a 11. ábrán látható.

$x_4 x_2$		$x_3 x_1$			
		01	11	01	00
01		x		x	x
11				x	
10				x	
00		x	x	x	x

11. ábra

### II.1.2. Határozatlan együtthatók módszere és a Harvard /minimizáló kártya/ módszer [2].

A határozatlan együtthatók módszere azon alapul, hogy bármely  $n$  változós függvény felírható

$$f(x_1, x_2, \dots, x_n) = \bigvee_{j=1}^n K_{i_1, i_2, \dots, i_j}^{\alpha_1, \alpha_2, \dots, \alpha_j} x_{i_1}^{\alpha_1} x_{i_2}^{\alpha_2} \dots x_{i_j}^{\alpha_j} \quad /6/$$

minden  $\alpha_1, \alpha_2, \dots, \alpha_j$  és  $i_1, i_2, \dots, i_j$ -re

alakban, ahol az együtthatók 0 vagy 1 értéket vehetnek fel. Behelyettesítve egy  $\alpha_1, \alpha_2, \dots, \alpha_n$  értékkombinációt a /6/ formulába,

$$f(\alpha_1, \alpha_2, \dots, \alpha_n) = \bigvee_{j=1}^n K_{i_1, i_2, \dots, i_j}^{\alpha_{i_1}, \alpha_{i_2}, \dots, \alpha_{i_j}} x_{i_1}^{\alpha_{i_1}} x_{i_2}^{\alpha_{i_2}} \dots x_{i_j}^{\alpha_{i_j}} \quad /7/$$

minden  $i_1, i_2, \dots, i_j$ -re és a megfelelő  $\alpha_{i_1}, \alpha_{i_2}, \dots, \alpha_{i_j}$ -re

alku formulát kapunk. Ha az  $f(\alpha_1, \alpha_2, \dots, \alpha_n) = 0$ , akkor a /7/ formulában előforduló együtthatók mindegyike 0. Véve az összes olyan  $\alpha_1, \alpha_2, \dots, \alpha_n$ -ér-



tékkombinációt, amelyekre  $f(\alpha_1, \alpha_2, \dots, \alpha_n) = 0$ , a /7/ alakú formuláknak egy olyan rendszerét kapjuk, amelyekben szereplő együtthatók megadják az  $f(x_1, x_2, \dots, x_n)$  /6/ alakú előállításának 0 értékű együtthatóit; a /6/ alakú előállításban szereplő többi együttható értéke 1.

### Eljárás:

Megadjuk az  $f(x_1, x_2, \dots, x_n)$  KDNF-jét. Minden  $\alpha_1, \alpha_2, \dots, \alpha_n$ -re felírjuk az  $f(\alpha_1, \alpha_2, \dots, \alpha_n)$ -ben szereplő együtthatókat és mindenütt töröljük az  $f(\alpha_1, \alpha_2, \dots, \alpha_n) = 0$ -hoz tartozó együtthatókat. Ily módon minden  $f(\alpha_1, \alpha_2, \dots, \alpha_n) = 1$ -ben megmarad együtthatóknak egy sorozata.

Kiválasztunk e sorozatokból együtthatóknak egy olyan halmazát, amely minden sorozatból tartalmaz legalább egy elemet, és amelyre a j-k összege minimális.

3. Példa:  $f(x_1, x_2, x_3) = x_1 x_2 x_3 \vee x_1 x_2 \bar{x}_3 \vee x_1 \bar{x}_2 x_3 \vee x_1 \bar{x}_2 \bar{x}_3 \vee \bar{x}_1 \bar{x}_2 \bar{x}_3$

Az  $\alpha_1, \alpha_2, \dots, \alpha_n$ -ekhez tartozó /7/ alakú formulák:

$$K_1^1 \vee K_2^1 \vee K_3^1 \vee K_{12}^{11} \vee K_{13}^{11} \vee K_{23}^{11} \vee K_{123}^{111} = f(1, 1, 1) = 1$$

$$K_1^1 \vee K_2^1 \vee K_3^0 \vee K_{12}^{11} \vee K_{13}^{10} \vee K_{23}^{10} \vee K_{123}^{110} = f(1, 1, 0) = 1$$

$$K_1^1 \vee K_2^0 \vee K_3^1 \vee K_{12}^{10} \vee K_{13}^{11} \vee K_{23}^{01} \vee K_{123}^{101} = f(1, 0, 1) = 1$$

$$K_1^1 \vee K_2^0 \vee K_3^0 \vee K_{12}^{10} \vee K_{13}^{10} \vee K_{23}^{00} \vee K_{123}^{100} = f(1, 0, 0) = 1$$

$$K_1^0 \vee K_2^1 \vee K_3^1 \vee K_{12}^{01} \vee K_{13}^{01} \vee K_{23}^{11} \vee K_{123}^{011} = f(0, 1, 1) = 0$$

$$K_1^0 \vee K_2^1 \vee K_3^0 \vee K_{12}^{01} \vee K_{13}^{00} \vee K_{23}^{10} \vee K_{123}^{010} = f(0, 1, 0) = 0$$

$$K_1^0 \vee K_2^0 \vee K_3^1 \vee K_{12}^{00} \vee K_{13}^{01} \vee K_{23}^{01} \vee K_{123}^{001} = f(0, 0, 1) = 0$$

$$K_1^0 \vee K_2^0 \vee K_3^0 \vee K_{12}^{00} \vee K_{13}^{00} \vee K_{23}^{00} \vee K_{123}^{000} = f(0, 0, 0) = 1$$

Az  $f(\alpha_1, \alpha_2, \dots, \alpha_n)$  értékeit figyelembe véve:  $K_1^0 = K_2^0 = K_3^1 = K_3^0 = K_3^1 =$   
 $= K_{12}^{00} = K_{12}^{01} = K_{13}^{00} = K_{13}^{01} = K_{23}^{01} = K_{23}^{10} = K_{23}^{11} = K_{123}^{001} = K_{123}^{010} = K_{123}^{011} = 0$

Ezek elhagyása után:

$$K_1^1 \vee K_{12}^{11} \vee K_{13}^{11} \vee K_{123}^{111} = 1$$

$$K_1^1 \vee K_{12}^{11} \vee K_{13}^{10} \vee K_{123}^{110} = 1$$

$$K_1^1 \vee K_{12}^{10} \vee K_{13}^{11} \vee K_{123}^{101} = 1$$

$$K_1^1 \vee K_{12}^{10} \vee K_{13}^{10} \vee K_{23}^{00} \vee K_{123}^{100} = 1$$

$$K_{23}^{00} \vee K_{123}^{000} = 1$$

Kiválasztva a feltételeknek megfelelő együttthatókat:

$$K_1^1 = 1$$

$$K_{23}^{00} = 1$$

Ennek alapján:  $f = x_1 \vee \bar{x}_2 \bar{x}_3$ .

Ugyanez a lényege a Harvard-módszernek [2] csak kézi használatra kényelmesebb. Adott egy előre elkészített táblázat, amelynek  $2^n$  sora és  $2^n$  oszlopa van. Egy sorban egy bizonyos minterm és az összes őt elnyelő  $i (i=1,2,\dots,n-1)$  változót tartalmazó elemi konjunkció szerepel.

- Eljárás:
1. Kihuzni minden, a függvényben nem szereplő minterm sorát és a többi sorokban szereplő, azokat az elemi konjunkciókat, amelyek a kihuzott sorokban szerepelnek.
  2. Megjelölni minden sorban a legkevesebb változóból álló elemi konjunkciókat. Ezzel az összes primimplikánst megjelöljük /néhányikat esetleg többször/.

A minimális alakok felírása ezekből a Quine-McCluskey algoritmusban szokásos módon történik.

4. Példa:  $f(x_1, x_2, x_3) = x_1 x_2 \bar{x}_3 \vee x_1 \bar{x}_2 x_3 \vee x_1 \bar{x}_2 \bar{x}_3 \vee \bar{x}_1 x_2 x_3 \vee \bar{x}_1 x_2 \bar{x}_3 \vee \bar{x}_1 \bar{x}_2 x_3$   
függvény alapján a táblázat a következőképpen alakul:



$x_1$	$x_2$	$x_3$	$x_1x_2$	$x_1x_3$	$x_2x_3$	$x_1x_2x_3$
<del><math>x_1</math></del>	<del><math>x_2</math></del>	<del><math>x_3</math></del>	<del><math>x_1x_2</math></del>	$(x_1\bar{x}_3)$	$(x_2\bar{x}_3)$	$x_1x_2\bar{x}_3$
<del><math>x_1</math></del>	<del><math>x_2</math></del>	<del><math>x_3</math></del>	$(x_1\bar{x}_2)$	<del><math>x_1x_3</math></del>	$(x_2x_3)$	$x_1\bar{x}_2x_3$
<del><math>x_1</math></del>	<del><math>x_2</math></del>	<del><math>x_3</math></del>	$(x_1x_2)$	$(x_1x_3)$	<del><math>x_2x_3</math></del>	$\bar{x}_1x_2x_3$
<del><math>x_1</math></del>	<del><math>x_2</math></del>	<del><math>x_3</math></del>	$(x_1\bar{x}_2)$	$(x_1\bar{x}_3)$	<del><math>x_2\bar{x}_3</math></del>	$x_1\bar{x}_2\bar{x}_3$
<del><math>x_1</math></del>	<del><math>x_2</math></del>	<del><math>x_3</math></del>	$(x_1x_2)$	<del><math>x_1x_3</math></del>	$(x_2x_3)$	$\bar{x}_1x_2x_3$
<del><math>x_1</math></del>	<del><math>x_2</math></del>	<del><math>x_3</math></del>	$(x_1\bar{x}_2)$	$(x_1\bar{x}_3)$	$(x_2\bar{x}_3)$	$\bar{x}_1\bar{x}_2\bar{x}_3$
<del><math>x_1</math></del>	<del><math>x_2</math></del>	<del><math>x_3</math></del>	<del><math>x_1x_2</math></del>	$(x_1x_3)$	$(x_2x_3)$	$\bar{x}_1x_2x_3$
<del><math>x_1</math></del>	<del><math>x_2</math></del>	<del><math>x_3</math></del>	<del><math>x_1x_2</math></del>	<del><math>x_1x_3</math></del>	<del><math>x_2x_3</math></del>	<del><math>x_1x_2x_3</math></del>

12. ábra

### II.1.3. Quine-McCluskey-féle algoritmus

A függvény KDNF-ja alapján előállítja a függvény MDNF-ját. Ezen algoritmussal kapcsolatos első publikáció [56] 1952-ben Quine, egy későbbi lényeges kiegészítés [19] 1956-ban McCluskey nevéhez fűződik.

Az algoritmus első része, az egyszerűsítési eljárás felírása két szerzőnél azonos és csak az  $ax \vee a\bar{x} = a$  összevonási és az  $ab \vee a = a$  elnyelési azonosságot használja fel.

Az összes primimplikáns felírása a következő lépésekben történik. Jelölje  $M$  az  $f$  mintermjeinek számát.

1. A  $M$  darab minterm listába rendezése.
2. Annak megvizsgálása minden  $i$ -re ( $i=1,2,\dots,M-1$ ), hogy a lista  $i$ -dik és  $k$ -dik ( $k > i$ ) eleme összevonható-e az  $/1/$  azonosság alkalmazásával. Ha igen, az érintett két elem megjelölése, az összevonás eredményeként kapott elemi konjunkciónak egy új listába történő elhelyezése, ha a



listában az illető elemi konjunkció még nem szerepel /elnyelés - /2/ azonosság/.

3. Ha a 2. lépés során új lista előállt /azaz a 2. lépés kiindulási listájában voltak összevonható elemek/, akkor az új listával a 2. lépés ismétlése, egyébként áttérés a 4. lépésre.
4. A részlistákon szereplő, de még nem jelölt elemi konjunkciók a primimplikánsok, - ezek alapján egy olyan lefedési táblázat készítése, amelynek sorai a primimplikánsoknak, oszlopai a mintermeknek felelnek meg. Az  $i$ -edik sor  $k$ -adik eleme 1, ha az  $i$ -edik primimplikáns elnyeli a  $k$ -adik mintermet, egyébként 0.

Az algoritmus második része Quine módszere szerint a lényeges primimplikánsok megkeresése.

Azok a primimplikánsok, amelyeknek megfelelő sorban az első rész során elkészített lefedési táblázat tartalmaz legalább egy olyan egyest, amelynek oszlopában nincs több egyes, lényeges primimplikánsok.

Az algoritmus szerint meg kell keresni ezeket; a sorokat és a táblázat azon oszlopait, ahol az elhagyott sorokban 1 állt, töröljük. Ezáltal a primimplikánsokra vonatkozó táblázatot kapunk, ahol csak a lényeges primimplikánsok által el nem nyelt mintermek szerepelnek. Töröljük ez után azokat a primimplikánsokat, amelyek sorában 1 nem szerepel - ezeket ugyanis elnyeli a lényeges primimplikánsok diszjunkciója /lényegtelen primimplikánsok/.

Az algoritmus eddigi menete egyértelmű. Az algoritmus harmadik része - a minimális lefedés keresése - a második rész befejezése után megmaradt primimplikánsok közül kiválasztja azokat, amelyek a lényeges primimplikánsokkal együtt a függvény egy MDNF-ját adják. Quine a második részben elkészített redukált táblázatnak kiválasztja a legtöbb egyest tartalmazó oszlopát és ezen oszlopbeli 1-ek sorai közül a legtöbb egyest tartalmazót. A megfelelő primimplikánst a lényeges primimplikánsok mellé veszi, a táblázatból elhagyja ezt a sort és az  $e$  sor egyeseit tartalmazó oszlopokat. Ezt az eljá-



rást addig folytatja, amíg a táblázat el nem fogy.

McCluskey, Quine intuitív megfontoláson alapuló kiválasztási algoritmusáról ellenpélda konstruálásával kimutatta [19], hogy az nem mindig adja meg a függvény MDNF-jét. McCluskey a Quine-algoritmus olyan módosítását javasolja, amely csaknem teljes kipróbálást igényel.

Az algoritmus második részében a lefedési táblázatban a primimplikánsokhoz Boole-változókat rendelve felírja a függvény primimplikánsokkal való összes lefedését megadó  $\Psi$  monoton Boole-függvényt, amely az egyes mintermeket elnyelő primimplikánsok diszjunkciónak konjunkciója. A disztributivitási szabály  $\Psi$ -re való alkalmazásával megkapja a  $\Psi$  függvény RDNF-ját. Az így kapott RDNF-beli bármely primimplikáns a függvény olyan primimplikánsait reprezentáló változókat tartalmazza, amelyeknek diszjunkciója előállítja a függvényt. Mivel e diszjunkciókat  $\Psi$  RDNF-ja alapján képeztük, ahol az egyes konjunkciókban szereplő változók száma minimális, a fenti előállítás a függvény IDNF-it - az összeset - adja. Az algoritmus harmadik részében a függvény összes MDNF-i közül kell - teljes végignézés alapján - kiválasztani a MDNF-t.

A McCluskey-algoritmus harmadik részének javítására több kísérlet történt, amelyek közül a legelterjedtebbek gráfelméleti eredményeket használnak fel. A minimális lefedés keresését legtöbbször gráfszínezési problémára vezetik vissza és keresik az összefüggést a gráf kromatikus száma és a MDNF-ban szereplő primimplikánsok száma között [26, 33, 49]. Ugyanehhez a problémához több szerző kapcsolódik, akik a teljes kipróbálás helyett a Boole-mátrixok minimális lefedésének módszereit keresik. Minimális lefedés a mátrix minimális számú olyan sorának kiválasztása, amelyekben lévő egyesek minden oszlopot fednek. Általános eredmény azonban még nem ismert.

McCluskey ezenkívül az egyszerűsítési eljárás során felhasznált mintermek és elemi konjunkciók egy olyan előrendezését [19] javasolja, amely az összevonható elemek keresését egyszerűsítheti. Az algoritmus ilyen finomítása



több munkában fellelhető. A cél általában az elemi konjunkciók konkrét reprezentációjától függő vagy független, az összevonhatóságot könnyen megállapító kritérium keresése [45]. Az algoritmus e változatainak kiinduló adata a függvény KDNF-ja. Mivel az összevonások során kapott elemi konjunkciókban egy változó vagy negálva vagy negálatlanul szerepel, vagy nem szerepel, az algoritmus lépései során az összes primimplikáns meghatározásához szükséges memóriakapacitás a változók számával,  $n$ -nel,  $3^n$  nagyságrendben nő. Így az algoritmus ismert gépi realizációjánál a változók száma erősen korlátozott, általában 8-10 lehet.

### 5. Példa:

Írjuk fel az alábbi mintermekkel adott függvény összes primimplikánsát a Quine-algoritmussal.

$$\bar{x}_1 \bar{x}_2 \bar{x}_3 x_4 \quad \checkmark$$

$$\bar{x}_1 \bar{x}_2 x_3 x_4 \quad \checkmark$$

$$\bar{x}_1 x_2 \bar{x}_3 \bar{x}_4 \quad \checkmark$$

$$\bar{x}_1 x_2 \bar{x}_3 x_4 \quad \checkmark$$

$$\bar{x}_1 x_2 x_3 x_4 \quad \checkmark$$

$$x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \quad \checkmark$$

$$x_1 \bar{x}_2 \bar{x}_3 x_4 \quad \checkmark$$

$$x_1 \bar{x}_2 x_3 x_4 \quad \checkmark$$

$$x_1 x_2 x_3 \bar{x}_4$$

$$x_1 x_2 x_3 x_4 \quad \checkmark$$

Az első összevonás utáni elemi konjunkciók:

$$\bar{x}_1 \bar{x}_2 x_4 \quad \checkmark$$

$$\bar{x}_1 \bar{x}_3 x_4 \quad \checkmark$$

$$\bar{x}_2 \bar{x}_3 x_4 \quad \checkmark$$

$$\bar{x}_1 x_3 x_4 \quad \checkmark$$

$$\bar{x}_2 x_3 x_4 \quad \checkmark$$

$$\bar{x}_1 x_2 \bar{x}_3$$

$$\bar{x}_1 x_2 x_4 \quad \checkmark$$



$$x_2 x_3 x_4 \checkmark$$

$$x_1 \bar{x}_2 \bar{x}_3$$

$$x_1 \bar{x}_2 x_4 \checkmark$$

$$x_1 x_3 x_4 \checkmark$$

$$x_1 x_2 x_3$$

A második összevonás utáni elemi konjunkciók:

$$\bar{x}_1 x_4$$

$$\bar{x}_2 x_4$$

$$x_3 x_4$$

A harmadik összevonási kísérlet új eredményt nem ad. A primimplikánsok a meg nem jelölt konjunkciók.

Igen gyakori a mintermek tízes számrendszerbeli számokkal történő olyan reprezentációja, amelynél a minterm a szám binárisan kódolt alakja /a változó negátalanul 1, negálva 0/. Ezzel a reprezentációval dolgozunk a 6. és 7. példában.

#### 6. Példa:

Legyen  $f$  a 0,3,4,5,6,7,8,10,11 mintermek diszjunkciója; az összevonások eredményeként kapott primimplikánsokat jelölje A,B,C,D,E,F,G.  $A=x_3 \bar{x}_4$ ;  $B=\bar{x}_1 \bar{x}_2 \bar{x}_4$ ,  $C=\bar{x}_1 \bar{x}_2 \bar{x}_3$ ,  $D=x_1 x_2 \bar{x}_4$ ,  $E=x_1 x_2 \bar{x}_3$ ,  $F=x_2 \bar{x}_3 x_4$ ,  $G=\bar{x}_1 \bar{x}_3 x_4$ .

	0	3	4	5	6	7	8	10	11
A			1	1	1	1			
B	1		1						
C	1						1		
D		1				1			
E		1							1
F								1	1
G							1	1	

13. ábra



A táblázat alapján a  $\Psi$  függvény:

$$\Psi = (B \vee C) (D \vee E) (A \vee B)(A)(A)(A \vee D) (C \vee G) (F \vee G) (E \vee F)$$

a kifejtés után

$$\Psi = ABDGF \vee ABEG \vee ACEG \vee ACFD \vee ACFE$$

Az  $f$  függvény IDNF-i:  $f = A \vee B \vee D \vee G \vee F$ ,  $f = A \vee C \vee F \vee D$ ,

$f = A \vee C \vee F \vee E$ ,  $f = A \vee B \vee E \vee G$ ,  $f = A \vee C \vee E \vee G$ ; ezek közül az első kivételével mind egyben MDNF is.

A Quine-McCluskey módszer egyik fontos általánosítása az általános Boole-függvények egyszerűsítésére szolgáló algoritmus. Ezzel a problémával különböző szerzők foglalkoztak [12, 17, 47], akik lényegében a Quine-McCluskey algoritmus felhasználásával értek el bizonyos eredményeket. A problémát azonban McCluskey és Schorr oldotta meg [17]-ben. Definíálják az együttes /több függvényhez tartozó/ primimplikánst /23. definíció/ és az együttes lényeges primimplikánst /24. definíció/. Az algoritmus kiinduló adata a mintermek listája, minden mintermnél indexként feltüntetve, hogy mely függvényekben szerepel. Az algoritmus lényegében megegyezik a Quine-algoritmus McCluskey-féle változatával, csak az első rész második lépésénél az összevonási feltételek bonyolódhatnak. Az  $ax \vee a\bar{x} = a$  összevonási azonosság azon azonos rangú, egy eltérésű konjunkciópárookra alkalmazható, amelyek indexhalmazainak metszete nem üres. Az összevonás eredményeként kapott konjunkció indexhalmaza az összevont mintermek indexhalmazainak közös része. Az  $ab \vee a = a$  elnyelési azonosság akkor alkalmazható, ha az  $a$  indexhalmaza része  $ab$  indexhalmazának; az eredmény indexhalmaza az " $a$ " konjunkció eredeti indexhalmaza marad. Az algoritmus első része így megadja a komponens függvényekre vonatkozó együttes primimplikánsokat. A kapott primimplikánsokból és az  $n$  függvény mintermjeiből táblázatot készítenek, ahol a sorok a primimplikánsoknak, az oszlopok az egyes függvények mintermjeinek felelnek meg, a közös mintermeket annyiszor véve, ahány függvényben előfordulnak.



## 7. Példa:

Legyen  $F = \{f_1, f_2, f_3\}$ , ahol:

$$f_1 = x_1 x_2 \bar{x}_3 \bar{x}_4 \vee x_1 x_2 x_3 \bar{x}_4 \vee x_1 x_2 \bar{x}_3 x_4 \vee x_1 x_2 x_3 x_4 \vee \bar{x}_1 \bar{x}_2 x_3 x_4$$

$$f_2 = x_1 x_2 \bar{x}_3 x_4 \vee x_1 x_2 x_3 x_4 \vee \bar{x}_1 x_2 \bar{x}_3 x_4 \vee \bar{x}_1 \bar{x}_2 x_3 x_4 \vee x_1 \bar{x}_2 x_3 x_4$$

$$f_3 = x_1 x_2 \bar{x}_3 \bar{x}_4 \vee x_1 x_2 x_3 \bar{x}_4 \vee x_1 \bar{x}_2 x_3 \bar{x}_4 \vee x_1 x_2 \bar{x}_3 x_4 \vee x_1 x_2 x_3 x_4 \vee \bar{x}_1 \bar{x}_2 x_3 x_4$$

A mintermek:

3	$\bar{x}_1 \bar{x}_2 x_3 x_4$	(1,2,3)
5	$\bar{x}_1 x_2 \bar{x}_3 x_4$	(2)
10	$x_1 \bar{x}_2 x_3 \bar{x}_4$	(3)
11	$x_1 \bar{x}_2 x_3 x_4$	(2)
12	$x_1 x_2 \bar{x}_3 \bar{x}_4$	(1,3)
13	$x_1 x_2 \bar{x}_3 x_4$	(1,2,3)
14	$x_1 x_2 x_3 \bar{x}_4$	(1,3)
15	$x_1 x_2 x_3 x_4$	(1,2,3)

A primimplikánsok:

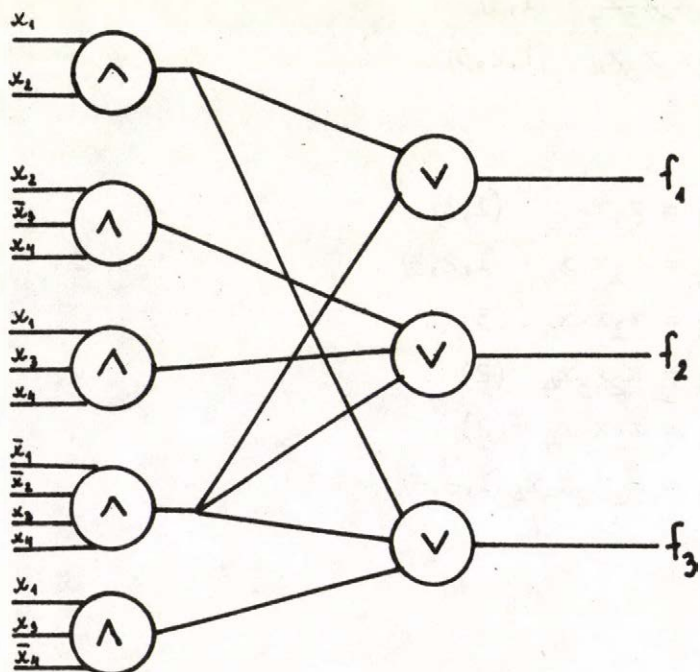
$$\begin{aligned} p_1 &= x_1 x_2 & (1,3) \\ p_2 &= x_1 x_2 x_4 & (1,2,3) \\ p_3 &= x_1 x_3 \bar{x}_4 & (3) \\ p_4 &= x_2 \bar{x}_3 x_4 & (2) \\ p_5 &= x_1 x_3 x_4 & (2) \\ p_6 &= \bar{x}_1 \bar{x}_2 x_3 x_4 & (1,2,3) \end{aligned}$$

A lefedési táblázat:

	1	2	3	4	5	6	7	8
	1 3	1 3	1 2 3	1 2 3	1 2 3	2 3	3 2	2
P <sub>1</sub>	+	+	+	+				
P <sub>2</sub>			+	+	+			
P <sub>3</sub>		+					+	
P <sub>4</sub>			+			+		
P <sub>5</sub>				+				+
P <sub>6</sub>					+	+	+	

14. ábra

Ez az eredmény a következő hálózattal szemléltethető:



15. ábra



#### II.1.4. Kipróbálás módszere

A kipróbálás módszerét először M.A. Gavrilov ismertette egy 1954-ben tartott előadásában, de csak később publikálta [27], mint McNaughton [47]. E módszer egy-egy gépi realizációjának leírása megtalálható [23]-ban és [48] ban.

A módszer azon a felismerésen alapul, hogy azok a konjunkciók, amelyek implikálják az  $f$  függvényt, de egyetlen változójuk elhagyása után már nem implikálják az  $f$ -et,  $e$  függvény primimplikánsai. Tehát rendre megvizsgálva az összes, egyetlen változót tartalmazó  $x_1^{\alpha_i}$  ( $i=1,2,\dots,n$ ) konjunkciót és megtartva közülük azokat, amelyek implikálják  $f$ -et, megkapjuk  $f$  összes egy rangu primimplikánsát. Majd megvizsgálva az összes olyan, két változót tartalmazó, kettő rangu  $x_1^{\alpha_i} x_j^{\alpha_j}$  konjunkciót, amelyeket  $f$  eddig megtalált primimplikánsai nem nyelnek el, ezek közül az  $f$ -et implikáló konjunkciók az  $f$  összes kettő rangu primimplikánsai. Általában a  $k$ -dik lépésben megvizsgálva az összes olyan  $k$  rangu  $x_1^{\alpha_{i_1}} x_2^{\alpha_{i_2}} \dots x_k^{\alpha_{i_k}}$  konjunkciót, amelyeket az előzőek során kapott primimplikánsok nem nyelnek el és megtartva közülük azokat, amelyek az  $f$ -et implikálják, megkapjuk az  $f$  összes  $k$  rangu primimplikánsait.

Az algoritmusban lényeges szerepet játszik annak eldöntési módja, hogy egy konjunkció implikálja-e a függvényt: implikáns az a konjunkció, amelyik nem nyel el 0 pontot /azaz neki megfelelő mintermet/, de elnyel legalább egy 1 pontot.

Gavrilov [27]-ben vizsgálta a  $\Phi$ -Boole-függvényeket is, amelyek igazságtábláját  $B_n$  azon pontjain adta meg, ahol a függvény meghatározott /azaz adottak 0 helyei és 1 helyei/. [23, 47, 48]-ban viszont kiinduló adat a függvény teljes igazságtáblája, amely minden helyre vonatkozólag tartalmazza azt, hogy ott a függvény értéke 1, 0 vagy meghatározatlan.

E módszer változatai tehát kiinduló adatként megkövetelik a Boole-függvény



igazságtáblájának megadását. Ez teljesen meghatározott Boole-függvények esetén az  $f$  és  $\bar{f}$  KDNF-jának ismeretét jelenti /ebben az esetben természetesen  $\bar{f}$  KDNF-ja  $f$  KDNF-jából is felírható/.  $\Phi$ -Boole-függvények esetén az igazságtábla megadása ekvivalens a függvény 0- és 1-helyei karakterisztikus függvényeinek,  $f_0$  és  $f_1$  KDNF-jának megadásával.

Egy függvény KDNF-jával történő megadása - a KDNF felírásának kényelmetlenségén kívül - általában komoly követelményeket /pl. nagy memóriakapacitás, nagy sebesség/ támaszt azzal a technikai berendezéssel szemben, amelyen az algoritmust alkalmazni kívánjuk.

Necula [48] a kipróbálás módszerét és a Ledley [38] könyvében lévő reprezentációs módszert továbbfejlesztve első lépésben kiválasztja az összes lényeges primimplikánst, majd - Gavrilovhoz hasonlóan - előállítja a függvény egy IDNF-ját.

Ha nem szükséges a MDNF előállítása, azaz elegendő egy IDNF meghatározása, akkor az iteráció befejeződik az olyan primimplikánsok megtalálása után, amelyek diszjunkciója ekvivalens az  $f$  függvénnyel. Ha viszont MDNF-át kell előállítani, az eljárás befejezésének kritériuma  $k=n$ .

II.1.5. A consensus-módszer előnye az összes eddig felsorolt módszerekkel szemben, hogy a függvény egy tetszőleges DNF-ja alapján írja fel annak összes primimplikánsát. Quine [57]-ben definiálja a consensus-fogalmat két elemi konjunkcióra: a  $\varphi$  és  $\psi$  konjunkciók  $\varphi\psi$  konjunkciójából a közös változók elhagyásával keletkező konjunkció az  $x\varphi$  és  $\bar{x}\psi$  konjunkciók consensusa.

Több szerző kimutatta, hogy a  $ax \vee b\bar{x} = ax \vee b\bar{x} \vee ab$  bővítési azonosság-nak és az elnyelési szabálynak az alkalmazásával megkapható az összes primimplikáns [54, 57]. Tison [61, 62, 63] munkáiban, amelyekről összefoglalót a [37] IV. fejezetében is találhatunk, kidolgozza a consensusok elméletét. Egyes lényeges definíciók és eredmények:



Egy  $\sigma$  elemi konjunkcióról azt mondjuk, hogy irredundáns módon része a  $\sigma_1$  elemi konjunkciók  $\bigvee \sigma_1$  diszjunkciójának, ha  $\sigma(\bigvee \sigma_i) = \sigma$ , de már egyetlen  $\sigma_1$  elhagyása után is  $\sigma(\bigvee' \sigma_i) \neq \sigma$

Könnyen belátható ez esetben a következő két tulajdonság:

1. a  $\sigma$ -ban szereplő változók egyetlen  $\sigma_1$ -ben sem szerepelnek más hatványon, mint  $\sigma$ -ban;
2. ha egy változó a  $\{\sigma_1\}$  halmazban csak negálva vagy csak negátlanul szerepel, akkor ez a változó ebben az alakjában szerepel  $\sigma$ -ban is.

$\sigma$  a  $\bigvee \sigma_i$  consensusa, ha  $\sigma$  irredundáns módon része  $\bigvee \sigma_i$ -nek és csak olyan változókat tartalmaz, amelyek legalább egy  $\sigma_1$ -ben előfordulnak.

A [37]-ben is megtalálható a következő tétel:

Legyen minden  $i$ -re  $\sigma_1$  egy-egy felbontása  $\mu_i \vee_i$ , ahol a  $\mu_i$  és  $\vee_i$  elemi konjunkciók nem tartalmaznak közös változókat. A  $\sigma = \prod_i \mu_i$  akkor és csak akkor consensusa  $\bigvee \sigma_1$ -nek, ha  $\bigvee \vee_i = 1$ .

Ebből speciálisan két elemi konjunkcióra adódik, hogy  $ax \vee b\bar{x}$  consensusa  $ab$ ; tehát a /4/-ben szereplő bővitmény a két diszjunkciós tag consensusa.

Egy másik fontos tétel: Ha  $p$  az  $f$  függvény primimplikánsa és  $\bigvee \sigma_i$  e függvény valamely DNF-ja, akkor van olyan  $\bigvee' \sigma_i$  részdiszjunkció, amelynek  $p$  consensusa [37. IV].

Erre a tételre alapozva Tison kidolgoz több algoritmust is a primimplikánsoknak a függvény diszjunktív normálformájából való felírására - az irredundáns, vagy minimális lefedés kérdésével nem foglalkozik. Ezen algoritmusok mindegyike az összes primimplikánst oly módon adja meg, hogy a  $\sigma_1$ -k részhalmazaira /esetleg kételemű részhalmazaira/ megkeresi a consensusot. E consensusok száma szinte előre nem becsülhető módon nő, ami indokolatlanul nagy memória és természetesen gépi idő igényhez vezet.



## II.1.6. Topológiai módszer

E módszer a Boole-függvények minimizálását átfogalmazza az  $n$  dimenziós Boole-kockában a függvény 1-pontjaira vonatkozó topológiai problémává, amelyet a Quine-algoritmusnak megfelelő módszerrel old meg.

E módszer előzményeként megemlíthető C.Y. Lee [30] cikke, bár a módszer kialakítása Urbano és Mueller [66] nevéhez fűződik.

Az  $n$  dimenziós Boole-kocka  $2^k$  pontból álló  $/k$  dimenziós/ részkockáját  $k$ -cellának nevezik. Ebben a megfogalmazásban egy  $l$  rangú konjunkciónak egy  $(n-l)$  -cella  $/2^{n-l}$  pontot lefedő intervallum/ felel meg.

A primimplikáns megfelelője a bázis cella, amely a függvény 1-pontjaiból álló olyan  $k$ -cella, amelyre  $k$  maximális. A lényeges primimplikánsnak a lényeges bázis cella felel meg. A lényeges bázis cellában van legalább egy olyan pont, amelyből pontosan annyi él indul ki, ahány dimenziós a lényeges bázis cella, mint Boole-kocka.

A függvény 1-pontjainak megfelelő összes csucst lefedő összes bázis cellák halmazát bázis cella-rendszernek nevezik.

A függvény egy pontjának megfelelő csucst tartalmazó, más szóval a csucshoz tartozó, bázis cellák halmazát bázis csillagnak nevezik. Lényeges bázis csillag az olyan bázis csillag, amely nem valódi részhalmaza egyetlen bázis csillagnak sem. Lényeges csucs az olyan csucs, amelyhez lényeges bázis csillag tartozik.

E fogalmak felhasználásával a Boole-függvény minimalizálása: az  $n$  dimenziós Boole-kockában a függvény 1-pontjainak a függvényhez tartozó bázis cellarendszer elemeivel való minimális lefedése.

Urbano és Mueller kimutatják, hogy:



- a./ minden, a függvényt lefedő cellahalmaz tartalmazza az összes lényeges bázis cellát;
- b./ a lényeges csucskok irredundáns lefedése /bármely bázis cella elhagyásával lesz le nem fedett csucs/ ekvivalens a függvényhez tartozó összes csucs irredundáns lefedésével.

Az algoritmus megkeresi a lényeges csucskokat, majd felírja a lényeges bázis csillagokat, végül kiválasztja a lényeges csucskok irredundáns vagy minimális lefedését.

A változószám növelésének lehetőségeivel is foglalkozik [66], amelyet a változók négyes csoportjainak vizsgálatával kíván elérni. A [36]-ban az eredeti módszer olyan átalakítását találjuk, amelyben az IDNF ill. MDNF keresése egyszerűbbé válik. Az eljárás lényege, hogy megkeresi az összes lényeges bázis cellát, megjelöli /már lefedett pontoknak tekinti/ azokat a pontokat, amelyeket lényeges bázis cella fed; ezekre nem kísérli meg bázis cella keresését. Ez az eljárás, mint megmutatják,  $\Phi$ -Boole-függvényekre is alkalmazható, ha a függvény azon pontjait, ahol a függvény nem meghatározott, már lefedett pontoknak tekintjük.

Az említett eljárások lényegéből adódóan kiinduló adatként az  $f$  /teljesen meghatározott/ vagy az  $\hat{f}$  KDNF-ját kell megadni. A  $\Phi$ -Boole-függvények /különösen a kevés helyen meghatározott függvények/ esetében azonban nem gazdaságos az  $\hat{f}$  KDNF-jának megadása. Ezenkívül a teljesen meghatározott függvényekre vonatkozó algoritmusoknak a  $\Phi$ -Boole-függvényekre történő mechanikus általánosítása nem képes kihasználni azt az információt, amelyet az  $f_0$  ismerete jelent. Az eljárásokban minden primimplikánst meghatározunk, hogy a minimális alak elérhető legyen, az összes primimplikáns felírása azonban igen bonyolult feladat. Ezért abban az esetben, ha elegendő egy IDNF-át megadni, érdemes egyszerűbb, esetleg nem az összes primimplikánst megadó algoritmust kidolgozni.



### II.1.7. Nelson tétele

Legyen adva  $f$  az  $f = \prod (x_{i_1}^{\alpha_{i_1}} \vee x_{i_2}^{\alpha_{i_2}} \vee \dots \vee x_{i_n}^{\alpha_{i_n}})$  konjunktív normálformában. Ha kifejtjük a kifejezést, azaz

alkalmazzuk a disztributív szabályt,  
elhagyjuk az azonosan nulla kifejezéseket, majd  
ahol lehet, alkalmazzuk az elnyelési azonosságot, megkapjuk  
a függvény RDNF-ját, amely a függvény összes primimplikánsá-  
nak diszjunkciója.

Tehát egy függvény összes primimplikánsa a függvény konjunktív normálformá-  
ja alapján rendkívül egyszerűen megkapható. /Itt technikai nehézséget a  
konjunktív normálforma felírása jelenthet./ A DNF-ban megadott függvények  
szorzására Voishvillo [68] dolgozott ki egy hatékony algoritmust.

### II.2. Egyes algoritmusok módosításai

E pontban az I. fejezetben kialakított szemléletmódunknak megfelelően meg-  
vizsgálunk a II.1.-ben ismertetett bizonyos algoritmusokat. E vizsgálat  
olyan változtatási lehetőségeket tár fel, amelyek alkalmasak az algoritmu-  
soknak a II.1. pontban említett hiányosságai kiküszöbölésére. Az algorit-  
musok ily módon kidolgozott új változatai kisebb követelményeket támaszta-  
nak a technikai berendezéssel szemben, vagy a minimálishoz közelebb álló  
eredményt adnak ill. általánosabban alkalmazhatók.

A továbbiakban tehát egy  $n$  változós Boole-függvényt úgy tekintünk, mint az  
 $n$  dimenziós Boole-tér pontjai egy  $R$  részhalmazának karakterisztikus függ-  
vényét. Egy minterm a tér egy pontjával /2. definíció/, egy  $k$  rangú elemi  
konjunkció a tér egy  $n-k$  dimenziós intervallumával ekvivalens /5. definíció/.  
A Boole-függvény primimplikánsainak megfelelői a  $B_n$  térben a függvény ka-  
rakterisztikus halmazának primintervallumai /10. definíció/.



A Quine-algoritmus összes primimplikánst meghatározó első részének a  $B_n$  tér pontjain végzett műveletekre történő átírása a függvény összes primintervallumának meghatározására vezet.

Szemléltetés kedvéért megadjuk a lépések átfogalmazását.

1. A függvény által lefedett  $R$  ponthalmaz pontjainak listába rendezése.
2. Az első iteráció: összevonási azonosság alkalmazása az  $R$  összes pontjának megfelelő elemi konjunkcióra. Ilyen értelemben két pont akkor és csak akkor "vonható össze", ha távolságuk 1. Eredmény a  $R$  összes, két pontból álló egydimenziós intervalluma. Azok a pontok, amelyek másokkal nem vonhatók össze, a  $R$  halmaz izolált pontjai - lényeges, 0 dimenziós primintervallumok. Az elnyelési azonosság megfelelőjének alkalmazása következtében elhagyjuk a többször szereplő intervallumokat és azokat a pontokat, amelyek valamely, az első iteráció során kapott intervallumnak pontjai.

Általában a  $k$ -adik iteráció az összevonási azonosságok alkalmazása a  $R$  összes  $n-(k-1)$  rangú intervallumára.

Összevonható bármely két,  $n-(k-1)$  rangú szomszédos /34. definíció/ és ugyanazon változókat tartalmazó intervallum. Eredmény  $R$  összes,  $2^k$  pontból álló,  $k$  dimenziós,  $n-k$  rangú intervalluma. Azok az intervallumok, amelyek nem vonhatók össze, a  $R$  halmaz egyetlen, nagyobb dimenziójú /kisebb rangú/ intervallumának sem részhalmazai, tehát primintervallumok. A  $R$  minden,  $k-1$  dimenziós intervallumára megnézve az összevonási azonosság alkalmazhatóságát, a  $k$ -adik lépésben megkapjuk az összes  $k-1$  dimenziós primintervallumot.

Az elnyelési azonosság alkalmazásának eredménye, a listában eredetileg többször szereplő  $k$  dimenziós intervallumok egyszer fognak szerepelni és törlődnek azok a  $k-1$  dimenziós intervallumok, amelyek nem primintervallumok.

3. A 2. lépés ismétlése mindaddig folytatódik, amíg az eredményeként előállt intervallum listában vannak összevonható elemek.



4. Az iteráció végeredményeként előállt lista az összes primintervallumok listája.

Ebből az átfogalmazásból látható, hogy a Boole-függvény összes primimplikánsainak meghatározása ekvivalens a függvény pontjaiból álló összes primintervallum - minimális rangú intervallum - megadásával.

Az összes primimplikáns felírásának problémáját ezáltal visszavezettük egy  $B_n$ -beli halmaz összes primintervallumai felírásának problémájára. Ez a probléma könnyebben kezelhető, mivel a primintervallumok meghatározása független a függvény megadási módjától, hiszen a függvény bármely formában történő megadása meghatározza a kiindulási halmazt.

Az alábbiakban ismertetjük a II.1.-ben leírt kipróbálás módszere és a consensus-módszer alapján kidolgozott algoritmusokat /II.2.1. ill. II.2.2./, amelyek megadják az összes primintervallumot. Ezen algoritmusok első fázisa a függvény megadott DNF-jának bizonyos egyszerűsítését végzi, ezt az egyszerűsített alakot előzetesen redukált DNF-nek nevezzük és ERDNF-fel jelöljük. A második fázis az 1. fázis eredményeként kapott ERDNF alapján előállítja a függvény RDNF-ját.

A felhasználók általában az eredményeket un. előkészített anyag formájában kérik, amely a következőket tartalmazza:

- a./ primimplikánsok /és lényeges primimplikánsok/ listái,
- b./ a primimplikánsokra és azon diszjunkt tartományokra vonatkozó /38. definíció/ lefedési táblázat, amelyeket a lényeges primimplikánsok egyike sem fed le.

A II.2.3. pontban ismertetünk egy eljárást a lényeges primimplikánsok kiválasztására /előkészített anyag a. része/, a II.2.4. pontban pedig módszert adunk a lefedési táblázat elkészítésére /előkészített anyag b. része/.



### II.2.1. Kipróbálás módszere

A függvény tetszőleges DNF-ban lehet adott. Ez egyrészt azt jelenti, hogy a kiinduló adatként a függvénynek a tervezési probléma szóbeli fogalmazása alapján vagy más úton természetesen adódó DNF-ja használható, másrészt, hogy csak e DNF elhelyezéséhez szükséges memóriakapacitást kell biztosítani.

A konkrét DNF-ból adódó lehetőségeket az iterációs lépések számának csökkentésére is felhasználhatjuk a következő megfontolás alapján. Ha a függvény adott DNF-jában szereplő konjunkciók rangjának minimuma  $r > 1$ , akkor célszerű a kipróbálást nem az 1, hanem az  $r$  rangú konjunkciókkal kezdeni, mivel lehet, hogy ennél kisebb rangú primimplikánsa nincs is a függvénynek.

Az alábbi tétel pedig az eljárás befejezésének egy olyan kritériumát nyújtja, amelynek felhasználásával a MDNF eléréséhez szükséges iterációs lépések száma tovább csökkenthető.

#### Tétel

Legyen  $f(x_1, x_2, \dots, x_n)$   $n$  változós Boole-függvény, továbbá  $\bigvee c_i = f$ , ahol minden egyes  $c_i$  egy-egy konjunkció és a  $c_i$ -k rangjának minimuma  $\ell$  a maximuma  $k$ , ( $\ell \leq k < n$ ). Tegyük fel még, hogy az összes  $\ell$ -nél magasabb rangú  $c_i$ -k primimplikánsok és az  $\ell$  rangú  $c_i$ -ken kívül nincs  $f$ -et implikáló  $\ell$  rangú konjunkció. Legyen  $q$  a  $c_i$  konjunkciók szomszédos párjaihoz tartozó bővitmények rangjának maximuma. Az  $f$ -nek akkor és csak akkor nem lesznek  $k$ -nál magasabb rangú primimplikánsai, ha nincs egyetlen olyan  $s$  rangú primimplikáns sem, amelyre  $k < s \leq q$ .

#### Bizonyítás

Egy függvény tetszőleges diszjunktív normálformájából a bővítési és az elnyelési azonosság alkalmazásával felírható a függvény összes primimplikánsa [55, 57].



Tegyük fel, hogy nincs  $s$  ( $k < s \leq q$ ) rangú primimplikáns. Ez azt jelenti, hogy a  $c_i$ -kből egyetlen bővitéssel kapható,  $k$ -nál magasabb rangú bővitményeket elnyeli valamelyik  $c_i$ . Megmutatjuk, hogy ebben az esetben  $q$ -nál magasabb rangú primimplikáns nem létezik.

Mivel  $q$ -nál magasabb rangú implikánst a  $c_i$ -kből csak bővitések sorozatával kaphatunk és mivel tetszőleges két szomszédos  $c_i$  és  $c_j$  bővitményét elnyeli valamelyik  $c_m$ , ezért valamely bővitéssorozat útján létrejövő bővitményt bármely olyan  $c_p$  elnyeli, amely elnyel e bővitménysorozatból legalább egyet. Így a  $q$ -nál magasabb rangú implikánsok egyike sem lehet primimplikáns. Ezzel a tételt bebizonyítottuk.

A kipróbálás módszerére kidolgozott új algoritmus a következő három típusú adatra alkalmazható.

- a./ adott a teljesen meghatározott függvény tetszőleges DNF-ja;
- b./ adott a teljesen meghatározott függvény és negáltja egy-egy tetszőleges DNF-ja;
- c./ adottak az  $f$   $\Phi$ -Boole-függvény 1- ill. 0-helyei halmazának karakterisztikus függvényei  $f_1$  és  $f_0$  tetszőleges DNF-ikkal.

A b./ ill. c./ esetben az algoritmus az  $f$  és  $\bar{f}$  ill.  $f_1$  és  $f_0$  összes primimplikánsait szolgáltatja egyidejűleg, mégpedig egyszerűbben, mintha az algoritmust külön-külön alkalmaznánk az  $f$ -re és  $\bar{f}$ -ra ill. az  $f_1$ -re és  $f_0$ -ra.

Ezek után az algoritmus, amely két fázisból áll, a következő.

#### Első fázis: ERDNF felírása

Jelölje  $r_{\min}$  az a./ és b./ esetben az  $f$ , a c./ esetben az  $f_1$  megadott DNF-jában diszjunkciós tagként szereplő intervallumok rangjának minimumát. Legyen továbbá  $K$  az  $f$  ill.  $f_1$ ,  $L$  pedig b./ és c./ esetben az  $\bar{f}$  ill.  $f_0$  primintervallumainak listája. Az  $a \Leftarrow b$  azt jelenti, hogy  $a$  felveszi a  $b$



értéket.

1.  $r \leftarrow r_{\min}$ ,  $q \leftarrow 0$ ,  $K$  ill.  $K$  és  $L$  üres.
2. Megvizsgáljuk az összes olyan  $r$  rangú intervallumot, amelyeket a  $K$  listában szereplő egyetlen alacsonyabb rangú intervallum se nyel el, és eldöntjük, hogy a vizsgált intervallum implikánsa-e  $f$ -nek /az a./ esetben/,  $f$ -nek vagy  $\bar{f}$ -nek ill.  $f_1$ -nek vagy  $f_0$ -nak /a b./ ill. c./ esetben/. Az  $f$  vagy  $f_1$  implikánsait a  $K$ ,  $\bar{f}$  vagy  $f_0$  implikánsait a  $L$  listába írjuk be.
3. Ha  $q=0$ , az akkor algoritmus a 4. lépéssel folytatódik, egyébként megnézzük, hogy az előző iterációs lépés adott-e új  $K$ -beli intervallumot. Ha igen, akkor az 5. lépéssel folytatjuk, egyébként a 6. lépéssel.
4. Ha a  $K$ -ban szereplő intervallumok által reprezentált konjunkciók diszjunkciója ekvivalens a függvénnyel, akkor az eljárást az 5. lépéssel, ha nem, a 7. lépéssel folytatjuk.
5.  $q$  értékének kiszámítása  $K$  alapján.
6. Ha  $q < r$ , akkor az algoritmust a 7. lépéssel, egyébként a 8. lépéssel folytatjuk.
7.  $r \leftarrow r+1$  és az eljárás a 2. lépéssel folytatódik.
8. az a./ esetben az eljárás befejeződik, a b./ és c./ esetben az  $L$  listához hozzávesszük  $\bar{f}$  ill.  $f_0$   $r$ -nél alacsonyabb rangú azon implikánsainak megfelelő intervallumokat, amelyeket egyetlen  $L$ -beli intervallum sem nyel el és az eljárás befejeződik.

Az eredményül kapott  $K$  listában  $r_{\min}$  és annál magasabb rangú intervallumok szerepelnek. A 2. lépés első iterációja biztosítja, hogy a  $K$  listabeli  $r_{\min}$  rangú intervallumok a függvény összes  $r_{\min}$  rangú implikánsát adják. A  $K$  listában szereplő többi intervallum a függvény összes  $r_{\min}$ -nél magasabb rangú primimplikáns. Hogy ezek az intervallumok primimplikánsoknak felelnek meg, ez könnyen belátható a következőképpen. A kiválasztás módja miatt ezen intervallumok egyike sem része nála kisebb, de  $r_{\min}$ -nél nem



kisebb rangú,  $f$ -et ill.  $f_1$ -et implikáló intervallumok egyikének sem. Mivel a  $K$  listában a függvény összes  $r_{\min}$  rangú implikánsa szerepel, a függvény egy, a  $K$  listabeli,  $r_{\min}$ -nél magasabb rangú implikánsát csak akkor nyelhetné el  $r_{\min}$ -nél alacsonyabb rangú implikáns, ha már  $r_{\min}$  rangú is elnyelné. Tehát a  $K$  listában szereplő,  $r_{\min}$ -nél nagyobb rangú implikánsok primimplikánsok. A tétel viszont biztosítja, hogy a listában az összes  $r_{\min}$ -nél magasabb rangú implikáns szerepel.

Azonban a függvénynek lehet  $r_{\min}$ -nél nem magasabb rangú primimplikánsa is; ezek megkeresését az algoritmus 2. fázisa végzi.

A fentiekből látható, hogy az első fázis eredményeként előálló lista talmilag ekvivalens a Quine-algoritmus első része 2. lépésének  $(n-r_{\min}-1)$ -dik iterációja után adódó listával. Az  $r_{\min}$ -nél nem magasabb rangú primimplikánsok tehát megkereshetők a Quine-algoritmus egyszerű alkalmazásával az 1. fázis eredményeként kapott intervallum lista  $r_{\min}$  rangú intervallumaira.

Bizonyos feltételek teljesülése esetén azonban meghatározott rangú primimplikánsok megkeresésére megadható a Quine-algoritmus 2. iterációs lépésénél gyorsabb módszer is. Ezért a 2. fázis a két módszer kombinációja.

#### Második fázis: RDNF felírása

1.  $r \leftarrow r_{\min}$ ,  $M$  az  $r$  rangú intervallumok listája
2.  $k$  az  $M$  lista  $r$  rangú elemeinek száma
3. Ha  $k \cdot r > \binom{k}{2}$ , akkor az eljárás a 4. lépéssel, egyébként a 8. lépéssel folytatódik
4. Az  $M$  lista  $r$  rangú elemeire elvégezzük az összes lehetséges összevonást az  $ax \vee a\bar{x} = a$  azonosság alapján. /Quine-algoritmus első rész, 2. lépés./
5. A lista  $r$  rangú elemeire és a kapott  $r-1$  rangú elemekre alkalmazzuk az elnyelési szabályt, ha lehet.



6. Ha az iteráció során előállt  $r-1$  rangú intervallum, akkor az eljárás a 7. lépéssel folytatódik, egyébként befejeződik.
7. Az  $r-1$  rangú intervallumokat a  $M$  listához írjuk.  
 $r \leftarrow r-1$ . Ha  $r \neq 1$ , akkor az eljárás a 2. lépéssel folytatódik, egyébként befejeződik.
8. A  $M$  lista  $r$  rangú elemeiből képezzük az összes olyan  $r-1$  rangú intervallumokat, amelyek elnyelnek  $M$ -beli  $r$  rangú intervallumot. Megtartjuk az  $f$ -et implikáló  $r-1$  rangú intervallumokat. Az eljárás az 5. lépéssel folytatódik.

A leírt algoritmusnak az a./ ill. a b./ és c./ esetre való alkalmazása csak a vizsgált intervallum implikáns voltának eldöntési módjában különbözik egymástól.

Az a./ esetben annak eldöntésére, hogy egy intervallum vagy konjunkció implikálja-e a függvényt, felhasználjuk azt a tényt, hogy ha az a intervallum része a  $b_1, b_2, \dots, b_k$  intervallumok által lefedett halmaznak, akkor  $\bigcup_{i=1}^k b_i = 1$ , ahol  $b_i$ -t  $b_i$ -ből a  $b_i$  és az a közös koordinátáinak elhagyásával kapjuk /lásd consensus-módszer/.

A b./ és c./ esetben egyszerűbb döntési eljárást is alkalmazhatunk. Ugyanis ha egy konjunkció implikánsa  $f$ -nek, akkor nem implikálhat  $\bar{f}$ -beli pontot. Tehát, ha  $\bar{f} = \bigvee c_i$ , akkor  $f$  egy tetszőleges a implikánsára  $\varphi(a, c_i) > 0$ . Ez a megállapítás nyilván igaz akkor is, ha  $f$  helyett  $f_1$ -et és  $\bar{f}$  helyett  $f_0$ -at mondunk. Ennek alapján a b./ és c./ esetben a döntési eljárás a következő. Ellenőrizzük, hogy van-e olyan konjunkció  $\bar{f}$  adott diszjunktív normálformájában, amely hasonló a vizsgált konjunkcióhoz. Ha ilyen nincs, a vizsgált konjunkció  $f$ -nek implikánsa. Ha ilyen van, akkor megnézzük, hogy van-e  $f$  adott diszjunktív normálformájában a vizsgált konjunkcióhoz hasonló konjunkció. Ha ilyen nincs, a konjunkció  $\bar{f}$ -nek implikánsa. Ha ilyen van, akkor a konjunkció nem implikánsa se  $f$ -nek, se  $\bar{f}$ -nek.

A most elmondottakon alapuló döntési eljárással tehát mind az  $f$ , mind az  $\bar{f}$  összes implikánsát egyidejűleg megkapjuk.



## II.2.2. Rövidítés módszere

A rövidítés módszerének első változata teljesen meghatározott Boole-függvényekre alkalmazható, de később  $\Phi$ -Boole-függvényekre is kiterjesztjük. Az algoritmus a consensus módszer elméleti megalapozásánál elért eredmények alapján épül fel, de attól nagymértékben eltér. Egyrészt az összes primimplikáns felírását két szakaszban végzi el úgy, hogy a részeredményeknek nem kell memóriakapacitást biztosítani; így gyakorlatilag csak annyi memóriára van szükség, amennyi a függvény adatainak és az algoritmus programjának tárolásához szükséges. Másrészt csak konjunkciópárokra vizsgálja a consensus létét.

Ezenkívül az általunk bevezetett eltérésfogalom /32. definíció/ alkalmazásával az algoritmus lépései nagymértékben egyszerűsödnek.

### Első fázis: ERDNF írása

A függvény adott diszjunktív normálformájában szereplő tagoknak /konjunkcióknak/ megfelelő intervallumokból kiindulva csak azokat a minimális rangú intervallumokat határozzuk meg, amelyek nem növelik az eddigi intervallumok számát.

Legyen  $K$  a függvény adott DNF-jában szereplő elemi konjunkcióknak megfelelő intervallumok listája,  $K(j)$ ,  $j=1,2,\dots,k$ , a lista  $j$ -dik eleme

1.  $k \leftarrow K$  elemszáma

$i \leftarrow 1$ .

2.  $l \leftarrow 1$

$j \leftarrow i+1$

3. Ha  $l > i$ , akkor az eljárás a 4., egyébként az 5. lépéssel folytatódik.

4.  $i \leftarrow i+1$ . Ha  $i=k$ , az eljárás befejeződik, egyébként a 2. lépéssel folytatjuk.

5. Ha  $K(l)$  és  $K(j)$  eltérése nulla, akkor a 6. lépéssel, ha egy, akkor a 7. lépéssel, egyébként a 11. lépéssel folytatódik az eljárás.



6. Ha az  $ab \vee a\bar{a}$  szabály alkalmazható, akkor az eljárás a 8. lépéssel, egyébként a 11. lépéssel folytatódik.
7. Ha az  $ax \vee a\bar{x}$  szabály alkalmazható, akkor a 8. lépéssel, egyébként a 9. lépéssel folytatjuk az eljárást.
8.  $K(l) \leftarrow K(l) \vee K(j)$   
 $K(j) \leftarrow K(k)$   
 $k \leftarrow k-1$   
 Ha  $j > k$ , akkor az eljárás befejeződik, egyébként a 2. lépéssel folytatódik.
9. Ha az  $acx \vee a\bar{x}$  szabály alkalmazható, akkor a 10. lépéssel, egyébként a 11. lépéssel folytatjuk az eljárást.
10. Az  $acx$ -nek megfelelő listaelem helyébe  $ac$  kerül. Az eljárásban a 4. lépés következik.
11.  $l \leftarrow l + 1$ . és a 3. lépés következik.

Az algoritmus befejezésekor kapott DNF egy ERDNF lesz. Ugyanis az így kapott intervallumok nem feltétlenül primintervallumok, hiszen a consensust az algoritmus végrehajtása során nem irtuk fel.

Például az

$$f = x_1\bar{x}_2\bar{x}_3 \vee \bar{x}_1x_2\bar{x}_3 \vee x_3x_4 \vee \bar{x}_1\bar{x}_2x_4 \vee x_1x_2\bar{x}_3x_4 \text{ formában}$$

adott függvénynél még a legkedvezőbb műveleti sorrend esetén is csak az  $x_1\bar{x}_2\bar{x}_3 \vee \bar{x}_1x_2\bar{x}_3 \vee x_3x_4 \vee \bar{x}_1\bar{x}_2x_4 \vee x_1x_4$  formulát kaphatjuk meg, holott a primimplikánsok  $x_1\bar{x}_2\bar{x}_3$ ,  $\bar{x}_1x_2\bar{x}_3$ ,  $x_4$ .

#### Második fázis: RDNF felírása

1.  $k \leftarrow$  a függvény ERDNF-jában szereplő konjunkcióknak megfelelő intervallumok száma;  
 $K$  ezen intervallumok listája;  
 $K(j)$ ,  $j=1,2,\dots,k$  a lista  $j$ . eleme.  
 $i \leftarrow 1$ .



2.  $l \leftarrow 1$ ;  
 $j \leftarrow i+1$ .
3. Ha  $l > i$ , akkor az eljárás a 4., egyébként az 5. lépéssel folytatódik.
4.  $i \leftarrow i+1$ . Ha  $i=k$ , akkor az eljárás befejeződik, egyébként a 2. lépéssel folytatódik.
5. Ha  $K(l)$  és  $K(j)$  eltérése egy, akkor a 6. lépéssel, egyébként a 10. lépéssel folytatjuk az eljárást.
6. Ha az  $abx \vee ac\bar{x}=abx \vee ac\bar{x} \vee abc$  szabály alkalmazható, akkor a 7., egyébként a 10. lépéssel folytatódik az eljárás.
7. Az  $abc$  consensus kiszámítása.
8. Ha van olyan  $m=1,2,\dots,j$ , hogy  $K(m)$ -re és  $abc$ -re alkalmazható az elnyelési szabály, akkor  $K(m) \leftarrow abc \vee K(m)$  és az eljárás a 10. lépéssel folytatódik, egyébként a 9. lépéssel.
9.  $k \leftarrow k+1$   
 $K(k) \leftarrow abc$
10.  $l \leftarrow l+1$ . Folytatás a 3. lépéssel.

Mivel egy függvény DNF-jából az elnyelési és bővítési szabály alkalmazásával a függvény RDNF-ja előáll [57], az algoritmus 2. fázisa a RDNF-át szolgáltatja.

A rövidítés előzőekben leírt módszere teljesen meghatározott Boole-függvények primimplikánsainak felírására alkalmas. Mint említettük ez az eljárás azonban csekély módosítással kiterjeszthető  $\Phi$ -Boole-függvények esetére is.

Legyen  $g(x_1, \dots, x_n)$  a  $B_n$  tér azon pontthalmazának karakterisztikus függvénye, ahol  $f$  nem definiált,  $f_1$  pedig azon  $B_n$  térbeli pontthalmazé, ahol  $f$  az 1 értéket veszi fel. Ekkor tetszőleges  $\gamma$  Boole-függvény mellett  $f=f_1 \vee \gamma g$  /lásd III.fejezet/.

A primimplikánsok felírása a következőképpen történik. Az előző algoritmus segítségével felírjuk  $f_1 \vee g$  összes primimplikánsát. Ezek közül azokat tartjuk meg, amelyek tartalmazznak  $f_1$ , más szóval  $f\bar{g}$ -beli pontot.



### II.2.3. Lényeges primimplikánsok kiválasztása

A 38. definícióval kapcsolatos tétel alapján az összes  $p_1$  primimplikáns ismeretében a függvény diszjunkt tartományait lefedő  $d_j = \prod_{i=1}^k p_i^{\alpha_i}$  ( $d_j \neq 0$ ,  $j=1,2,\dots,2^k-1$ ) Boole-függvények segítségével a lényeges primimplikánsok könnyen megkereshetők. Ugyanis lényeges primimplikáns minden olyan  $p_1$  primimplikáns, amelyhez van olyan  $d_j \neq 0$ , amelynek fenti formulájában csak  $p_1$  szerepel negátlanul, mivel a  $d_j$  által lefedett pontokat csak  $p_1$  fedi és mivel – a definíció miatt – az összes, csak  $p_1$  által lefedett pont pontja  $d_j$ -nek.

$\Phi$ -Boole-függvények esetén a lényeges primimplikánsok kiválasztása  $d_j$  helyett  $d_j^* = d_j \bar{g}$  függvények segítségével történik.

Az eljárás a következő.

Felírjuk azt a  $k$  darab  $d_j = \prod_{i=1}^k p_i^{\alpha_i}$ -t  $j=1,2,\dots,k$ / vagy  $d_j^*$ -t, ahol

$$\alpha_j = \begin{cases} 1 & \text{ha } i=j \\ 0 & \text{egyébként} \end{cases}$$

A  $d_j \neq 0$  ( $d_j^* \neq 0$ ) esetben  $p_j$  lényeges primimplikáns.

### II.2.4. A lefedési táblázat felírása

Legyenek  $p_1, p_2, \dots, p_s$  a függvény lényeges primimplikánsai. A táblázat oszlopaihoz rendelt diszjunkt halmazok a  $d_1 = \bar{p}_1 \bar{p}_2 \dots \bar{p}_s \prod_{j=1}^s p_j^{\alpha_j}$ , ill.

$d_1 = \bar{g} \bar{p}_1 \bar{p}_2 \dots \bar{p}_s \prod_{j=1}^s p_j^{\alpha_j}$  ( $d_1 \neq 0$ ,  $d_1^* \neq 0$ ,  $\exists_j (j \in [s+1, k] \wedge \alpha_j = 1)$ ). A táblázat sorai az egyes /nem lényeges/ primimplikánsokhoz tartoznak. A táblázat  $\tau_{ji}$  eleme 1, ha a  $d_1$  ill.  $d_1^*$  fenti alakjában  $\alpha_j = 1$ , egyébként  $\tau_{ji} = 0$ . A táblázat előállítási módja a következő:

1. Felírjuk az összes  $d_1 = \bar{p}_1 \bar{p}_2 \dots \bar{p}_s \prod_{j=1}^s p_j^{\alpha_j}$  ill.  $d_1^* = \bar{g} \bar{p}_1 \bar{p}_2 \dots \bar{p}_s \prod_{j=1}^s p_j^{\alpha_j}$ -t.
2. A  $\tau_{ji}$  értékek kiszámítása és beírása.



### II.3. Gráf-módszer

A II.1. pontban ismertetett eljárások közül a topológiai módszer illik bele leginkább az I. fejezetben kialakított szemléletbe. Amint a módszer ismertetésénél rámutattunk, az eddig kidolgozott változatok komoly hiányosságokkal rendelkeznek. Az e pontban ismertetésre kerülő gráf-módszer a topológiai módszer olyan általánosítása, amelyhez a topológiai módszernek új szempontok alapján történő vizsgálatával jutottunk és, amely új, hatékony algoritmusok kidolgozásához vezetett. A kidolgozás során egyik célunk a topológiai módszer alapvető hiányosságainak kiküszöbölése és az általános Boole-függvények szintézisének megoldása volt. Másik cél az előkészített anyaghoz szükséges információnak a primimplikánsok meghatározása során könnyen használható formában való kialakítása volt. Ez a célkitűzés eddig egyetlen minimizáló eljárásnál sem merült fel, bár ez az új szempont az algoritmusok hatékonyságának növeléséhez vezet. Az alább ismertetésre kerülő négy algoritmusban ezt a célt úgy érjük el, hogy a függvény pontjaihoz és primimplikánsaihoz indexként rendeljük azt az információt, amely az eljárások befejeződésekor lényegében az előkészített anyagot adja.

A gráf-módszer alapján kidolgozott első algoritmus teljesen meghatározott Boole-függvények összes primimplikánsának meghatározására szolgál. Ez az algoritmus a [66]-ban ismertetett algoritmus olyan módosítása, amely az információ választott ábrázolásából adódó lehetőségeket használja ki az eljárás gyorsítására.

A második algoritmus nem szolgáltatja az összes primimplikánst, hanem a teljesen meghatározott Boole-függvényt lefedő valamely primimplikáns halmazt eredményez.

A harmadik algoritmus  $\Phi$ -Boole-függvényekre vonatkozik. Ezen algoritmus végrehajtása során használt információ tartalma a  $\Phi$ -Boole-függvények természetéhez igazodik és az a [69]-ben leírt algoritmus lényegesen egyszerűbb



és gyorsabb változatát eredményezi.

A negyedik algoritmus a harmadik általánosítása általános Boole-függvények esetére.

Ezen algoritmusok kidolgozásához az a probléma vezetett, hogy átvihetők-e az automaták elméletében az automaták belső állapotai számának minimalizálására kidolgozott eljárások a Boole-függvények DNF-jának minimalizálására. E probléma vizsgálatával adódó algoritmusok elemzése megmutatta, hogy az algoritmusok alapjául szolgáló módszer a topológiai módszer továbbfejlesztéseként is felfogható.

A Boole-függvények DNF-jának minimalizálása és a nem teljesen meghatározott automaták állapotai számának minimalizálása közötti analógia első pillanatra szembeötlő. Ugyanis mindkét esetben az első feladat egy adott halmaz elemeinek az elemek bizonyos tulajdonságai alapján egyértelműen meghatározott, maximális méretű, nem feltétlenül diszjunkt részhalmazokba történő besorolása; a második feladat pedig az adott halmaz minimális vagy nem redundáns lefedése részhalmazai segítségével. Tekintve, hogy a második feladat Boole-függvények esetén számunkra az előkészített anyag megadását jelenti, a továbbiakban csak az első feladattal foglalkozunk részletesen.

Az adott halmaz az automaták esetében az állapotok halmaza, a Boole-függvények esetében pedig a függvényt meghatározó  $B_n$  térbeli pontok halmaza. Automaták esetén az állapotok részhalmazokba /kompatibilis osztályokba/ történő besorolását meghatározó tulajdonság az átmenet- és kimenetfüggvények értékeivel kapcsolatos; Boole-függvények esetén pedig a  $B_n$  tér pontjainak besorolását a primimplikánsokhoz való tartozás tulajdonsága határozza meg.

Az automaták állapotminimalizálására vonatkozó eljárások [7,8,15,18,52] első része - az összes maximális kompatibilis osztály meghatározása - lényegében a következő lépésekből áll:



- a./ Az átmenet- és kimenetfüggvények ismeretében meghatározzuk az összes kompatibilis állapotpárt;
- b./ Felírjuk az automata összeférhetőségi gráfját, amelynek csucsai az automata állapotainak felelnek meg, és két csucsát akkor és csak akkor köti össze él, ha azok kompatibilis párt alkotnak;
- c./ Kiválasztjuk az összeférhetőségi gráf összes maximális teljes részgráfját - e részgráfok csucsainak megfelelő állapotok halmazai alkotják a maximális kompatibilis állapothalmazokat.

Nyilvánvaló, hogyha egy Boole-függvény pontjaira a pontok primimplikánsokhoz való tartozása alapján értelmezzük az összeférhetőségi gráfot, akkor az automatakra kidolgozott fenti algoritmusnak a Boole-függvény ezen összeférhetőségi gráfjára történő alkalmazása az összes primimplikánst adná. Az ilyen irányú vizsgálatok azonban azt mutatták, hogy a Boole-függvény összeférhetőségi gráfjának a felírása szinte bonyolultabb feladat lenne, mint maguknak a primimplikánsoknak a meghatározása. Ezért olyan eljárásokat dolgoztunk ki, amelyek a Boole-függvény összeférhetőségi gráfjának megadását megkerülik.

A továbbiakban tehát egy adott Boole-függvény összeférhetőségi gráfján értjük azt a gráfot, amelynek csucsai a függvény pontjai és két csucsot akkor és csak akkor köt össze él, ha a függvénynek van olyan primimplikánsa, amely a két csucsot fedi. Ez azt jelenti, hogy a függvény összeférhetőségi gráfjának maximális teljes részgráfjain lévő - kompatibilis - pontok a függvény primimplikánsainak pontjai.

A Boole-függvény Boole-gráfjának nevezzük azt a gráfot, amelynek csucsai a függvény pontjai, de csak a szomszédos pontokat köti össze él. Nyilvánvaló, hogy a függvény Boole-gráfja a függvény összeférhetőségi gráfjának részgráfja. A Boole-függvény szomszédos pontjai ugyanis kompatibilis pontpárok, mivel mindig van legalább egy olyan primimplikáns, amelynek ezek pontjai, viszont nem minden egy primimplikánshoz tartozó pontpár szomszédos. Mint



majd belátjuk, a függvény összeférhetőségi gráfjának egy teljes részgráfján lévő pontjai a függvény Boole-gráfjában bizonyos meghatározott típusú utakkal össze vannak kötve. Éppen ezért, a primimplikánsok megkereséséhez elegendő a függvény Boole-gráfjának ismerete, ha ezt az információt kiegészítjük az utak kiválasztásának /a gráf bejárásának/ olyan szabályával, amely az egy primimplikánshoz tartozó összes pont elérését biztosítja. Ilyen szabály adódik az egy primimplikánshoz tartozó pontok koordinátái közötti összefüggések alapján. Ez a szabály viszont olyan formai tulajdonságokat tár fel, amelyek a tulajdonképpeni bejárást feleslegessé teszik.

Az algoritmusok elvi megalapozását a II.3.1. pontban a teljesen meghatározott Boole-függvényekre végezzük el.

### II.3.1. Teljesen meghatározott Boole-függvények vizsgálata

Legyen adott egy  $f$  Boole-függvény. Ha az  $f$  két pontja közti relációként fogjuk fel azt a tényt, hogy köztük az  $f$  Boole-gráfjában létezik él /azaz szomszédosak/, akkor ez a reláció szimmetrikus, reflexív és intranzitív.

A függvény pontjai között értelmezhetünk ekvivalencia relációt is a következőképpen. Legyen  $\{p_1, p_2, \dots, p_k\}$  az  $f$  primimplikánsainak egy, az  $f$  függvényt lefedő halmaza. Az  $f$  függvény  $P_i$  és  $P_j$  pontja akkor és csak akkor van e relációban egymással, ha van olyan  $\{p_{\ell_1}, p_{\ell_2}, \dots, p_{\ell_m}\} \subseteq \{p_1, p_2, \dots, p_k\}$ , hogy mind  $P_i$ , mind  $P_j$  pontjai minden  $p_{\ell_r}$ -nek ( $r=1, 2, \dots, m$ ) és nem pontja a  $\{p_1, p_2, \dots, p_k\} \setminus \{p_{\ell_1}, p_{\ell_2}, \dots, p_{\ell_m}\}$  egyetlen elemének sem. E definíció alapján kapott ekvivalenciaosztályok az  $f$  Boole-függvény összes megadott  $p_1, p_2, \dots, p_k$  primimplikánsára vonatkozó diszjunkt tartományait adják /38. definíció/.

Az  $f$  függvény pontjai közötti ezen ekvivalencia és kompatibilitás definíciója alapján érthető, hogy az algoritmusokban a primimplikánsok meghatározása során a diszjunkt halmazokra vonatkozó információ egyszerűen adódik.



A következőkben megadjuk az algoritmusokban felhasznált néhány fogalom definícióját és ismertetjük, esetleg igazoljuk a fontosabb összefüggéseket.

Azt mondjuk, hogy a  $B_n$  tér egy  $P$  pontjának a  $P'$  pont szomszédja az  $i$ -edik koordináta szerint, ha a  $P$  és  $P'$  pontok pontosan az  $i$ -edik koordinátában különböznek egymástól.

Az  $f$  függvény egy  $P$  pontjának van szomszédja az  $i$ -edik koordináta szerint a függvény pontjainak halmazában, ha van az  $f$  pontjai között olyan  $P'$  pont, amely a  $P$ -nek szomszédja az  $i$ -edik koordináta szerint.

### Konjunkciók vizsgálata

Mint az ismert, egy  $p = x_{i_1}^{\alpha_{i_1}} x_{i_2}^{\alpha_{i_2}} \dots x_{i_k}^{\alpha_{i_k}}$  konjunkció  $2^{n-k}$  pontot fed le. E pontok jellemzője, hogy  $i_1, i_2, \dots, i_k$  koordinátáik megegyeznek, a többi  $n-k$  koordináta pedig bármely lehetséges értékkombináció lehet. Ennek fordítottja is igaz: ha adott  $2^{n-k}$  olyan pont, amelyek  $i_1, i_2, \dots, i_k$  koordinátái megegyeznek, és ezek  $\alpha_{i_1}, \alpha_{i_2}, \dots, \alpha_{i_k}$  akkor a  $p = x_{i_1}^{\alpha_{i_1}} x_{i_2}^{\alpha_{i_2}} \dots x_{i_k}^{\alpha_{i_k}}$  konjunkció az adott pontokat – és csak azokat – fedi. Tehát egy konjunkció által lefedett bármely  $P$  pontot megadó mintermből elhagyva azt az  $n-k$  változót, amelyek  $P$ -vel szomszédos pontokhoz vezetnek, a konjunkciót kapjuk. Ezt a konjunkciót a  $P$ -re illeszkedő /vagy  $P$ -re létező/ konjunkciónak is nevezzük. Amennyiben az  $n-k$  koordinátának lényeges szerepe van, akkor kihangsúlyozzuk, hogy a  $P$  pontra ezen  $n-k$  koordináta szerint illeszkedő konjunkcióról van szó. Az előzőkből az is következik, hogy a  $p$  konjunkció által lefedett bármely  $P$  pont azon  $n-k$  koordinátája szerinti szomszédai, amely koordináták nem szerepelnek a  $p$  konjunkcióban, a konjunkciónak pontjai.

A  $p = x_{i_1}^{\alpha_{i_1}} x_{i_2}^{\alpha_{i_2}} x_{i_3}^{\alpha_{i_3}} \dots x_{i_k}^{\alpha_{i_k}}$  konjunkció által lefedett pontokra, koordinátáik speciális alakja és számuk meghatározottságán kívül, bizonyos távolsági összefüggések is érvényesek. Mivel ezen pontok egymástól csak a nem közös  $n-k$  koordinátában különbözhetnek, a  $p$  által lefedett bármely két pont



távolsága legfeljebb  $n-k$ . Nyilvánvaló, hogy a  $p$  által lefedett pontok halmazában bármely pontból még a tőle legtávolabbi  $n-k$  távolságra lévő pontba is el lehet jutni az  $n-k$  koordináta egymásutáni, tetszőleges sorrendben történő megváltoztatásával.

Ha a  $p_1, p_2, \dots, p_t$  konjunkcióknak egynél több közös pontjuk van, e közös pontok azon koordinátái, amelyeknek megfelelő változók a konjunkciók közül legalább egyben szerepelnek, egyformák.

### Implikánsok és primimplikánsok elemzése

Egy konjunkció akkor és csak akkor implikánsa egy  $f$  függvénynek, ha a konjunkció minden pontja a függvénynek is pontja. Egy konjunkció akkor és csak akkor primimplikánsa a függvénynek, ha változóinak száma  $k$ , minimális, vagyis  $n-k$ -nál több koordináta változtatásával kapott pontok között lesznek a függvényhez nem tartozók is.

Legyen  $P$  az  $f$  Boole-függvény egy pontja. Jelöljük  $s$ -sel az  $f$  függvény  $P$ -vel szomszédos pontjainak számát,  $K$ -val azon koordináták halmazát, amelyek szerint  $P$ -nek az  $f$  függvény pontjainak halmazában szomszédai vannak. Tehát  $|K| = s$ . Legyen  $p$  az  $f$  függvény egy primimplikánsa.

Az elmondottakból következik, hogy:

- Ha a  $p$  primimplikáns izolált, akkor minden pontjának pontosan  $n-k$  szomszédja van a függvény pontjai között. A primimplikáns minden pontjához ugyanaz a  $K$  tartozik és a  $K$  halmaz elemei azon változók indexei, amelyek nem szerepelnek a primimplikánst megadó konjunkcióban.
- Ha a  $2^{n-k}$  pontot fedő  $p$  primimplikáns lényeges primimplikáns, akkor kell legalább egy olyan pontjának lennie, amelyre  $s=n-k$ . Ezt a pontot /vagy pontokat/ csak a  $p$  lényeges primimplikáns fedi.
- Ha a  $2^{n-k}$  pontot fedő  $p$  primimplikáns nem lényeges primimplikáns, akkor



bármely pontjára  $s > n-k$ , de a primimplikánsban nem szereplő változóknak megfelelő koordináták halmaza valódi részhalmaza a  $p$  által fedett bármely  $P$ -hez tartozó  $K$ -nak.

#### Tétel:

Az azonosan 1-től különböző bármely teljesen meghatározott Boole-függvény minden primimplikánsának van legalább egy olyan pontja, amelyre  $s < n$ .

#### Bizonyítás:

Tegyük fel, hogy a függvény minden pontjában  $s=n$ . Ez azt jelenti, hogy a függvény bármely pontjából egy távolságra lévő minden pont pontja a függvénynek. Ez azonban csak akkor teljesülhet, ha a  $B_n$  tér minden pontja pontja a függvénynek, azaz a függvény az azonosan 1 függvény.

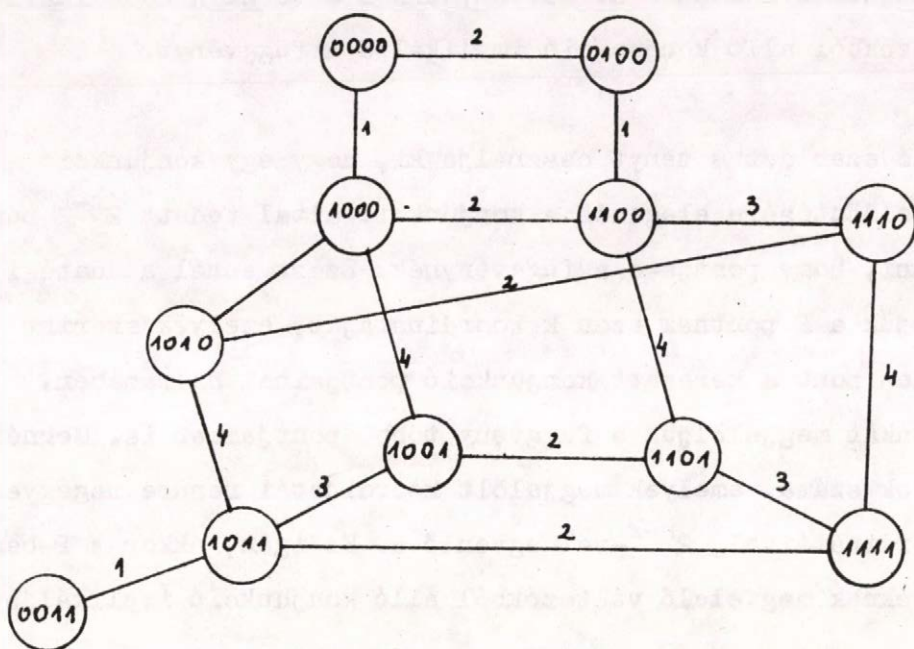
#### Következmény:

Az összes primimplikáns felírásához elegendő az  $s < n$  számú szomszédal rendelkező pontokat megvizsgálni.

Egyszerűen belátható, hogy ha egy Boole-függvény pontjainak száma  $L$ , ahol  $2^1 \leq L < 2^{1+1}$  és a minimális szomszédszám  $s_{\min} > 1$ , az illető Boole-függvénynek nincs lényeges primimplikánsa.

Az előzőek alapján tehát a primimplikánsok felírásához elegendő, ha a függvény minden pontjához megadjuk szomszédai  $s$  számát és azon koordináták  $K$  halmazát, amelyek szerint a pontnak szomszédai vannak a függvény pontjainak halmazában. Ez az információ megadható a Boole-gráf segítségével úgy, hogy az egyes élekhez az összekötött szomszédos pontokban eltérő koordináta sor-számát, a csucsokhoz pedig az illető pont koordinátáit és szomszédai számát írjuk. Az így kapott gráfot adatgráfnak nevezzük /16. ábra/.





Az  $f = a \vee \bar{c}d \vee \bar{b}cd$  Boole-függvény adatgráfja

16. ábra

#### Adott P pontra illeszkedő konjunkciók vizsgálata

Annak eldöntésére, hogy a függvény egy P pontjára illeszkedő konjunkció implikáns-e, két módszert adunk meg.

Az első módszerhez az a tény vezet, hogyha a P pontra illeszkedő konjunkció implikánsa az  $f$  függvénynek, akkor a konjunkció által fedett bármely P pontnak a konjunkcióban nem szereplő változóknak megfelelő koordináták mindegyike szerint biztosan van szomszédja az  $f$  függvény pontjainak halmazában. Tehát, ha az  $f$  függvény adott P pontjába illeszkedő, a függvényt implikáló konjunkciót keresünk, akkor a következőképpen járhatunk el. Vesszük a P pont azon  $n-k$  koordinátáját, amely szerint szomszédai vannak a keresett konjunkció pontjainak halmazában. Bejárjuk a gráfnak a P pontból az  $n-k$  koordinátával megjelölt éleken elérhető részét és megvizsgáljuk, hogy minden elért szögpontnak van-e ezen  $n-k$  koordináta mindegyike szerinti



szomszédja az  $f$  pontjainak halmazában. Ha van, akkor a többi  $k$  koordinátának megfelelő változókból álló konjunkció implikálja a függvényt.

A második döntési módszer azt a tényt használja ki, hogy egy konjunkció implikáns voltának eldöntésére elegendő a konjunkció által fedett  $2^{n-k}$  pont mindegyikére megnézni, hogy pontja-e a függvénynek. Ezért ennél a döntési módszernél megjelöljük a  $P$  pontnak azon  $k$  koordinátáját, amelyek szerint nincs vele szomszédos pont a keresett konjunkció pontjainak halmazában. Ugyanezen koordinátákat megjelöljük a függvény többi pontjaiban is. Megnézzük, hogy azon pontok száma, amelyek megjelölt koordinátái rendre megegyeznek  $P$  megjelölt koordinátaival,  $2^{n-k}$ -val egyenlő-e. Ha igen, akkor a  $P$ -ben megjelölt koordinátáknak megfelelő változókból álló konjunkció implikálja a függvényt.

Ha az  $f$  függvény valamely  $P$  pontjára  $s=n-k$  és a  $P$ -re ezen  $n-k$  koordináta szerint illeszkedő konjunkció implikáns, akkor ez lényeges primimplikáns. Ha a  $P$  pontra az  $n-k$  koordináta szerint illeszkedő konjunkció nem implikáns, akkor a  $P$ -re egynél több primimplikáns illeszkedik. Ebben az esetben a feladat a  $P$  pontra illeszkedő összes primimplikáns felírása. Ebből a célból a szomszédos pontokhoz vezető  $n-k$  koordináta közül ki kell választani azon  $n-k-i$  ( $i=1,2,\dots,n-k-1$ ) elemű koordináta-csoportokat, amelyek szerint a  $P$ -re illeszkedő konjunkciók primimplikánsok.

A kiválasztás módja a topológiai módszernél a következő. Sorra képezzük az  $1,2,\dots,n-k$  számok összes  $n-k-i$ -ed osztályu kombinációit ( $i=1,2,\dots,n-k-1$ ), ezek szerint kiválasztják a koordinátákat és döntenek, hogy a  $P$ -re a kiválasztott koordináták szerint illeszkedő konjunkció implikáns-e, majd, hogy primimplikáns-e.

A fenti módszernél sokkal gazdaságosabb eljárást dolgoztunk ki a kombinációk előállítására, amelyet az alábbiakban ismertetünk.

Ezen új eljárás azon kombinációk nagy részét, amelyeknek megfelelő koordi-



náták szerint a P-re illeszkedő konjunkciót valamely, már megtalált primimplikáns elnyelne, nem is állítja elő.

Könnyű belátni, hogy ha az  $\ell$ -1-ed osztályu kombinációkból úgy képezzük az  $\ell$ -ed osztályuakat, hogy mindegyikhez hozzávesszük egyenként a bennük szereplő legnagyobb számnál nagyobb számokat, akkor az összes  $\ell$ -ed osztályu kombinációt megkapjuk és mindegyiket csak egyszer.

Ezt a tényt használjuk fel a P-re illeszkedő konjunkciók kiválasztásánál, ahol a szomszédszám csökkentését növekvő osztályu kombinációk alapján végezzük. Vesszünk egy C kombinációt és ehhez kiválasztjuk a P-re illeszkedő azon konjunkciót, amely a C-ben nem szereplő koordináták szerint illeszkedik P-re. Ha a  $C'$  a C-nél magasabb osztályu olyan kombináció, amelyben C minden eleme szerepel, akkor a C-hez kiválasztásra kerülő konjunkciót a C-hez kiválasztott konjunkció elnyeli. Ebből következik, hogyha a kombinációk képzése során eljutunk egy olyan C kombinációhoz, amelyhez kiválasztott konjunkció primimplikáns, akkor a C-vel a fenti relációban álló C kombinációkhoz kiválasztott konjunkciókat a C-hez kiválasztott primimplikáns elnyelné, tehát a C-k képzése szükségtelen.

#### Az eljárás a következő:

Legyen a P pontnak s szomszédja. Tegyük fel, hogy a szomszédok számát minimalisan j-vel kell csökkenteni. Jelölje  $k_y$  a képzett kombináció osztályát,  $n_v$  a képzés során kapott v-ed osztályu kombinációk számát,  $NI(i)$  a v-ed osztályu kombinációkat ( $i=1,2,\dots,n_v$ ),  $Nl(i)$  az első osztályu kombinációkat ( $i=1,2,\dots,s$ ).

1.  $k_y \Leftarrow 0$ ,  $n_v \Leftarrow 1$ ,  $NI\ 1 \Leftarrow 0$ ,  $Nl(i) \Leftarrow 1$  ( $i=1,2,\dots,s$ )

2.  $n_w \Leftarrow 0$ ,  $k_y \Leftarrow k_y + 1$ ,  $v \Leftarrow k_y$

3.  $m \Leftarrow 0$

4.  $m \Leftarrow m + 1$

5. Ha  $NI(m) < Nl(v)$ , akkor a 6. lépés, egyébként a 9. lépés következik.



6.  $NI(m)$ -ből és  $NI(v)$ -ből képezzük a  $ky$ -ad osztályu  $C$  kombinációt
7. Ha  $ky < j$ , akkor a 14. lépéssel folytatjuk az eljárást.
8.  $nw \Leftarrow nw+1$   
 $NI1(nw) \Leftarrow C$
9. Ha  $m \neq nw$ , akkor a 4. lépés következik
10. Ha  $v=s$ , akkor  $v \Leftarrow v+1$  és a 3. lépés következik
11. Ha  $nw=0$ , akkor az eljárás befejeződik, egyébként  
 $NI(l) \Leftarrow NI1(l), l=1,2,\dots,nw, nv \Leftarrow nw$
12. Ha  $ky=1$ , akkor  $NI(k) \Leftarrow NI(k), k=1,2,\dots,nw; s \Leftarrow nw$
13. Ha  $ky=s$ , akkor az eljárás befejeződik, egyébként a 2. lépéssel folytatódik
14. Ha  $C$ -t a további kombinációk képzéséhez meg kell tartani, akkor a 8. lépéssel folytatódik az eljárás
15. A 9. lépés következik.

A fenti eljárást az e pontban ismertetésre kerülő gráf-módszer alapján működő algoritmusokban fogjuk alkalmazni minden olyan esetben, amikor egy pontra illeszkedő több primimplikánst keresünk.

### Ekvivalencia osztályok meghatározása

Az algoritmusok céljául tűztük ki azonban mint már említettük - a primimplikánsok felírásain kívül - az ekvivalencia osztályok /diszjunkt tartományok/ meghatározását is. A továbbiakban röviden ezzel a problémával foglalkozunk.

Az ekvivalencia osztályok meghatározásának egyik módja a következő: a megtalált primimplikáns minden pontjához hozzárendelünk egy a primimplikánssra utaló indexet. Ily módon, amikor a primimplikánsok felírása befejeződik, az ekvivalencia osztályok is kialakulnak, ugyanis azok és csak azok a pontok tartoznak egy ekvivalencia osztályba, amelyek indexei azonosak.

E módszer alkalmazása esetén egyrészt a primimplikánsok felírását még egy, az azonos indexű pontokat összegyűjtő lépésnek kell követnie, másrészt a



primimplikánsok felírása során nem hagyhatók figyelmen kívül a továbbiakban már szerepet nem játszó pontok. Az eljárás e két hátrányának kiküszöbölésére célszerű lenne az ekvivalens pontok halmazainak kiválasztására egy olyan módszer, amely a számítás során folyamatosan leválasztaná a továbbiakban figyelmen kívül hagyható pontokat. Ehhez szükséges lenne a primimplikánsok felírása során azon pontok ismerete, amelyekre minden primimplikánst felírtunk már, azaz, amelyek indexeinek halmaza véglegesen kialakult. E pontok felismerésére abban az esetben, ha az éppen felírt primimplikáns lényeges primimplikáns, a következő egyszerű kritérium adódik.

Az  $f$  függvény egy lényeges primimplikánsának felírásakor az illető primimplikáns által lefedett pontok közül azoknak és csak azoknak az indexhalmaza alakult ki véglegesen, amelyekhez tartozó  $K$  halmaz azonos azzal a koordinátahalmazzal, amelynek megfelelő változók nem szerepelnek a felírt konjunkcióban.

E kritériumnak eleget tevő pontok tehát a lényeges primimplikáns felírásakor megadnak bizonyos ekvivalencia osztályokat és a primimplikánskeresés további vizsgálatából kizárhatók - elhagyható pontok. Az  $f$  függvény egy  $p'$  lényeges primimplikánsának felírásakor elhagyva ezeket a pontokat, olyan  $f'$  függvényhez jutunk, amelynek primimplikánsai  $f$  primimplikánsai az éppen felírt  $p'$  kivételével.  $f'$  lényeges primimplikánsai egyrészt  $f$   $p'$ -től különböző lényeges primimplikánsai, másrészt  $f$  azon primimplikánsai, amelyeknek csak a  $p'$ -vel van közös pontjuk - ezek az  $f$  függvénynek  $p'$  elhagyásával lényegessé váló primimplikánsai.

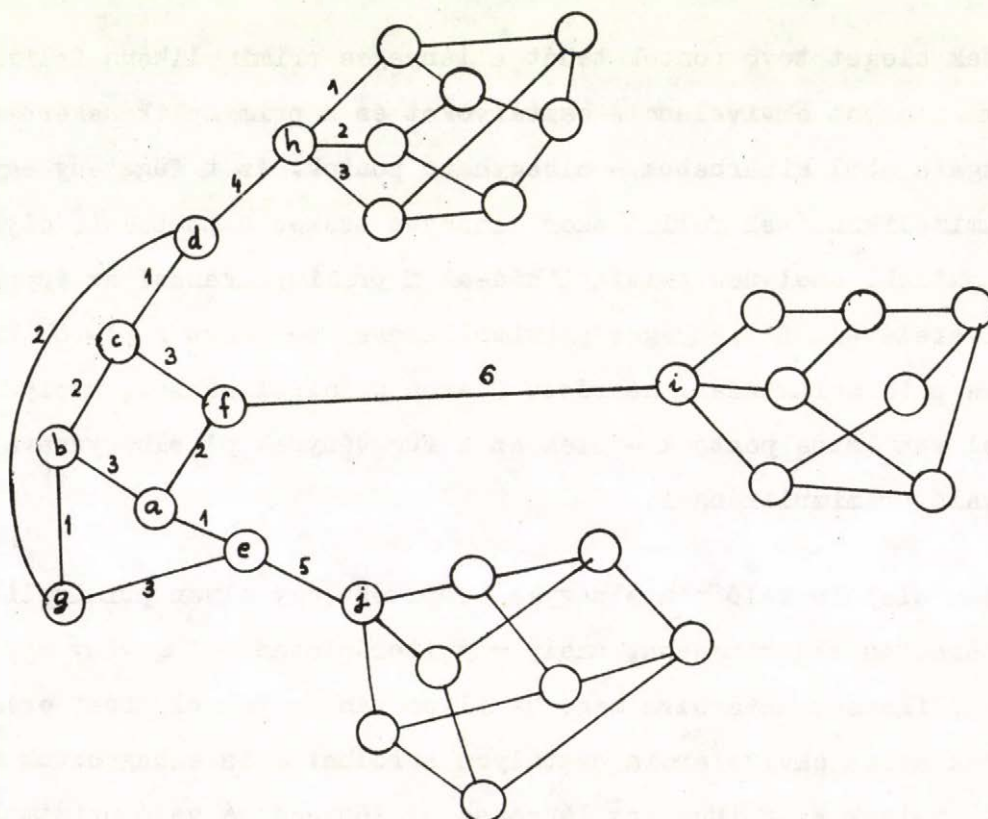
Az  $e$  kritérium alapján való pontelhagyás beépítése egy olyan primimplikáns előállító iterációs algoritmusba, amely egy iterációban a függvény egy lényeges primimplikánsát határozza meg, ha ilyen van, olyan eljárást eredményez, amelynek során ekvivalencia osztályba sorolhatók és elhagyhatók mindazon pontok, amelyek az  $f$  függvény lényeges és lényegessé váló primimplikánsai felírásakor eleget tesznek a kritériumnak. Ezen iterációs algoritmus



végeredményeként előálló függvénynek nincs lényeges primimplikánsa, tehát a kritérium alapján tovább nem kezelhető. Ezért célszerű lenne a kritérium általánosítása erre az esetre. Ekkor a lényeges primimplikáns szerepét egy olyan  $p_1, p_2, \dots, p_s$  primimplikánshalmaz játszhatná, amelyhez van legalább egy olyan pontja a függvénynek, amelyet ezek és csak ezek a primimplikánsok fednek. A kritérium természetes általánosítása lehetne ezek után a következő tulajdonság. A  $p_1, p_2, \dots, p_s$  mindegyike által lefedett pontok közül elhagyhatók azok és csak azok a pontok, amelyekhez tartozó  $K$  halmaz azonos azzal a koordinátahalmazzal, amelyeknek megfelelő változók a  $p_1, p_2, \dots, p_s$  primimplikánsok közül legalább egyben nem szerepelnek.

Azonban azt, hogy ez a tulajdonság nem alkalmas az elhagyható pontok meghatározására, a következő példán szemléltetjük.

Legyen a függvény Boole-gráfja a következő:



17. ábra



Válasszuk ki az  $a$  pontot. Az  $a$ -ra illeszkedő primimplikánsok  $p_1$  és  $p_2$ . A  $p_1$  az  $a, b, c, f$ , a  $p_2$  pedig az  $a, b, g, e$  pontokat fedi. Annak ellenére, hogy mind az  $a$ , mind a  $b$  mindkét primimplikánsnak pontja és a hozzájuk tartozó  $K$  is azonos,  $p_1$  és  $p_2$  elhagyásakor csak az  $a$  pont hagyható el, mivel  $b$ -re még egy primimplikáns illeszkedik /amely a  $b, c, d, g$  pontokat fedi/.

E példa alapján látható, hogy a fentebb megfogalmazott tulajdonság nem alkalmas az elhagyható pontok meghatározására.

Ez a Boole-gráf azonban arra is példa, hogy abban az esetben, amikor a függvénynek nincs lényeges primimplikánsa, nem feltétlenül gazdaságos minden elhagyható pont elhagyása. Ez a tény feleslegessé teszi a fent megfogalmazott tulajdonságnál bonyolultabb kritérium keresését az elhagyható pontok meghatározására. E megfontolás alapján abban az esetben, amikor a függvénynek nincs lényeges primimplikánsa, nem feltétlenül zárjuk ki a figyelmen kívül hagyható pontokat.

#### A primimplikánsok felírására szolgáló algoritmusok

Az algoritmusokhoz szükséges kiinduló adat a függvény adatgráfja.

A primimplikánsokat a függvény egyes pontjaira bizonyos - szomszédos pontokhoz vezető - koordináták szerint illeszkedő konjunkciók formájában keressük.

A konjunkciók felírásához a függvény pontjait szomszédszámuk növekvő sorrendje szerint járjuk be. Ez azért előnyös, mert egyrészt a kisebb szomszédszámú pontoknál kevesebb a variálási lehetőség még akkor is, ha nem illeszkedik lényeges primimplikáns erre a pontra, másrészt mivel  $s = \log_2 L$  -nél / $L$  a függvény pontjainak száma/ több szomszéddal rendelkező pontokra nem létezik primimplikáns - a lényeges primimplikánsoknak az elsők közötti megtalálása biztosítva van.

Az előző pontokban elmondottak alapján figyelembe véve a lényeges primimp-



likánsok felírásakor elhagyható pontokra a megadott kritériumot, az összes primimplikáns felírására egy két részből álló algoritmus látszik megfelelőnek. Az algoritmus első része felírja az összes lényeges és lényegessé váló primimplikánst úgy, hogy minden iterációban sorra megvizsgálja az előző iterációs lépésben kapott adatgráf /az első iterációban az eredeti adatgráf/ pontjait, mindaddig, amíg nem talál olyan pontot, amelyre lényeges primimplikáns illeszkedik. Minden egyes lényeges primimplikáns felírásakor elhagyja az adatgráfból a kritériumnak eleget tevő /elhagyható/ pontokat. Az algoritmus második része az összes lényeges és lényegessé váló primimplikáns elhagyása után kapott adatgráfon dolgozik. Az algoritmus egy iterációban a sorrendben következő P pontra illeszkedő összes primimplikánst adja meg és a P pontot a következő iterációkban figyelmen kívül hagyhatónak deklarálja. Ezt az algoritmust az alábbiakban részletezzük.

Jelölje  $L$  a függvény pontjainak számát,  $P_i$  az  $i$ -edik pontot,  $s_i$  ill.  $K_i$  az  $i$ -edik pont szomszédainak számát ill. szomszédaihoz vezető koordináták halmazát,  $n$  az  $f$  függvény változóinak számát.

#### A algoritmus

##### 1. rész

1. A függvény pontjait szomszédaik számának növekvő sorrendje szerint rendezzük:  $i \leftarrow 0$ .
2.  $i \leftarrow i+1$
3. Ha  $s_i > \lceil \log_2 L \rceil$ , akkor áttérés a 2. részre.
4.  $P_i$ -re  $K_i$  szerint illeszkedő  $p_i$  konjunkció felírása, azaz  $p_i$ -nek a  $K_i$ -ben nem szereplő koordinátáinak megjelölése.
5. Ha  $p_i$  nem implikánsa  $f$ -nek, akkor az algoritmus a 2. lépéssel folytatódik.
6.  $p_i$  által lefedett pontokat a  $p_i$ -re utaló indexszel látjuk el.  
E pontok közül elhagyjuk a kritériumnak eleget tevő, azaz az elhagyható pontokat: megfelelő csucok és élek törlése a gráfból,  $L$  csökkentése.



7. Ha  $L = 0$ , akkor az algoritmus az 1. lépéssel folytatódik.
8. Az algoritmus befejeződik, mivel már az első rész megadja az összes primimplikánst.

## 2. rész

1.  $i \Leftarrow 0$
2.  $i \Leftarrow i+1$
3. Ha  $s_i = n$ , akkor az eljárás befejeződik.
4. Ha az  $1, 2, \dots, n$  számok  $k$ -ad osztályu kombinációt előállító algoritmusunk szerint nem lehet képezni  $C \subset K_1$ -t, akkor az eljárás a 8. lépéssel folytatódik.
5.  $C$  kiválasztása  
 $P_1$ -re  $C$  szerint illeszkedő  $p$  konjunkció felírása.
6. Ha  $p \not\rightarrow f$  vagy  $p$  az előző iterációs lépések során felírásra került, akkor az eljárás a 3. lépéssel folytatódik.
7.  $p$  pontjainak  $p$ -re utaló indexszel való ellátása. Az eljárás a 4. lépéssel folytatódik.
8. A  $P_1$  pontot megjelöljük /a későbbiekben figyelmen kívül hagyható/.
9. Ha  $L \neq i$ , akkor az eljárás a 2. lépéssel folytatódik, egyébként az eljárás befejeződik.

Az A algoritmus megadja az összes primimplikánst, azonban az algoritmus mindkét része igen munkaigényes. Az első részben, annak ellenére, hogy minden iterációban csökkentjük a pontok számát, egy-egy lényeges primimplikáns felírásához általában az adatgráf több pontját is be kell járni. A második részben az egyes  $P_1$  pontokra illeszkedő minden primimplikáns felírása a pontok megadott sorrendjében történik, tehát a gráf  $s_1 < n$  szomszédszámu pontjait egyszer kell bejárunk. Mivel az algoritmus végrehajtása során nincs mód annak eldöntésére, hogy minden primimplikánst felírtunk-e már, az összes primimplikáns meghatározásához az egyszeri bejárást akkor is el kell végezni, ha az összes primimplikáns felírása már közben



megtörtént. Ebben az esetben egy primimplikáns többször is felírásra kerülhet, tehát biztosítani kell a többszörösség kiküszöbölését.

Miután a legtöbb esetben elegendő valamely IDNF felírása, tehát nem szükséges meghatározni az összes primimplikánst, kidolgoztunk egy olyan algoritmust, amely az A algoritmusnál gazdaságosabb. Ez az algoritmus a gráf-módszer alapján meghatároz valamely, a függvény pontjait általában többszörösen lefedő primimplikáns halmazt. Ez az algoritmus a bejárás során a függvény minden pontját, azokat is, amelyeknek szomszédszáma  $n$ , legfeljebb egyszer érinti; a figyelembe veendő pontokat növekvő szomszédszámuk sorrendjében dolgozza fel. Egy iterációs lépésben felírja a soronkövetkező  $P$  pontra a  $P$ -hez tartozó  $K$  szerint illeszkedő konjunkciót. Ha ez implikánsa a függvénynek, akkor lényeges primimplikáns. Ha ez a konjunkció nem implikánsa a függvénynek, akkor az algoritmus felírja a  $P$  pontra illeszkedő összes primimplikánst. Mindkét esetben fel nem dolgozandónak deklarálja azokat a pontokat, amelyeket legalább egy, éppen felírt primimplikáns fed.

#### B algoritmus

1. A függvény pontjainak szomszédszámuk növekvő sorrendje szerinti rendezése  
 $i \leftarrow 0$   
 $m \leftarrow L$
2.  $i \leftarrow i+1$
3. Ha  $P_i$  nem feldolgozandó, akkor áttérés a 2. lépésre.
4. Ha  $s_i = n$ , akkor a 11. lépéssel folytatódik az algoritmus
5. Felírjuk  $P_i$ -re a  $K_i$  szerint illeszkedő  $p$  konjunkciót
6. Ha  $p$  nem implikánsa  $f$ -nek, akkor áttérés a 11. lépésre.
7.  $p$  pontjait ellátjuk a  $p$ -re utaló indexszel és a figyelmen kívül hagyhatóság jelével  
 $m \leftarrow$  feldolgozandó pontok száma
8. Ha  $p$  nem lényeges primimplikáns, akkor áttérés a 11. lépésre.



9. Ha  $m=0$ , akkor az algoritmus befejeződik.
10. Ha  $i=m$ , akkor az algoritmus befejeződik, egyébként a 2. lépéssel folytatódik.
11. Ha  $K_1$ -ből már nem lehet primimplikáns előállítására alkalmas  $C$  koordinátahalmazt kiválasztani, akkor az eljárás a 9. lépéssel folytatódik.
12.  $C$  kiválasztása  
 $P_1$ -re  $C$  szerint illeszkedő  $p$  konjunkció felírása. Áttérés a 6. lépésre.

### II.3.2. $\Phi$ -Boole-függvények minimalizálása

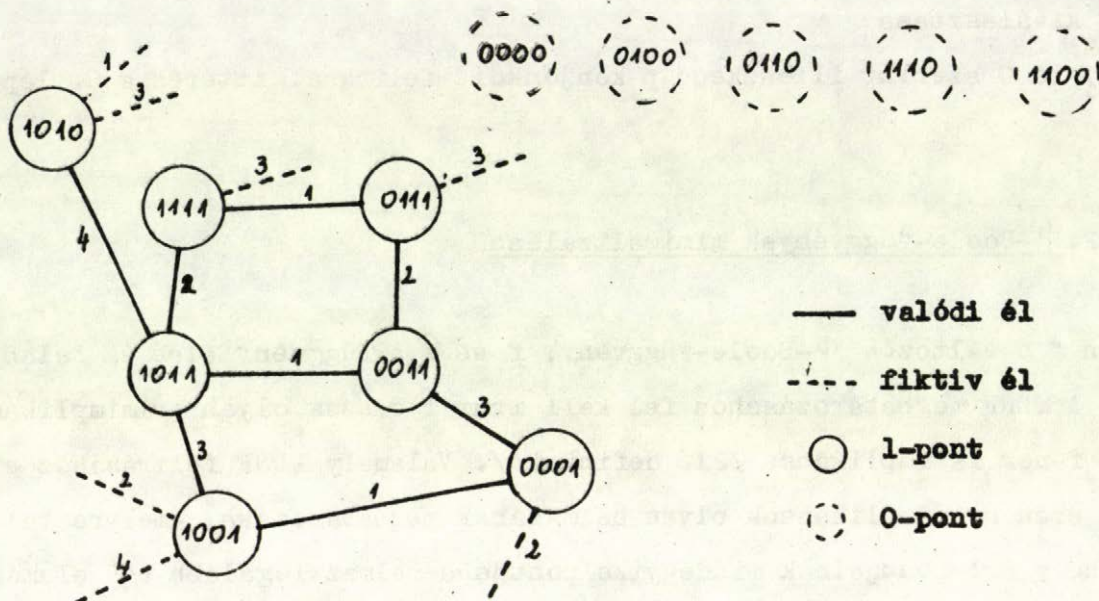
Legyen  $f$   $n$  változós  $\Phi$ -Boole-függvény;  $\underline{f}$  és  $\hat{f}$  a függvény alsó és felső határa. A MDNF meghatározásához fel kell írni  $\hat{f}$  összes olyan primimplikánsát, amely  $f$ -nek is implikánsa /21. definíció/. Valamely IDNF felírásához elegendő ezen primimplikánsok olyan halmazának meghatározása, amelyre teljesül, hogy  $f$  1-pontjainak mindegyike pontja a halmaz legalább egy elemének. A teljesen meghatározott függvényekre megadott módszer alkalmas  $\Phi$ -Boole-függvények primimplikánsainak felírására is. Az eljárás ebben az esetben az  $\hat{f}$  Boole-gráfja alapján felírja az  $f$  pontjaira illeszkedő összes primimplikánst, vagy ezen primimplikánsok olyan  $\{p_i\}$  részhalmazát ( $i=1,2,\dots,k$ ), amelyre  $f \rightarrow \bigvee_{i=1}^k p_i$  fennáll.

A probléma ilyen megoldása azonban, különösen a kevés helyen meghatározott függvények esetén, sok felesleges adat megadását követeli meg. Ezért az eljárás változatlanul tartása mellett az adatgráf új változatát vezetjük be, amely alkalmas a  $\Phi$ -Boole-függvények leírására és a lehető legkevesebb adat tárolását követeli meg.

Az adatgráf  $\Phi$ -Boole-függvényekre vonatkozó általánosítása a következőképpen definiált  $\Phi$ -adatgráf. A gráf szögpontjainak halmaza két diszjunkt halmaz egyesítése. Az egyik halmaz az  $\underline{f}$ , a másik az  $f_0$  pontjainak halmaza. Ez utóbbi halmaz elemei a gráf izolált pontjai. A gráf éleit két csoportba osztjuk, valódi és fiktív élekre. A valódi élek az  $\underline{f}$  függvény szomszédos



pontjait kötik össze, az egy adott pontra illeszkedő fiktív élek pedig azt mutatják, hogy az adott pontnak van szomszédos pontja  $\hat{f}$ -ben. A gráf éleihez minőségüktől függetlenül hozzárendeljük annak a változónak megfelelő koordinátát, amelyben a szomszédos pont eltér az adott ponttól /lásd 18. ábra./



Az  $f = x_2 x_3 x_4 \vee \bar{x}_2 x_4 \vee x_1 \bar{x}_2 x_3$ ,  $f_0 = x_2 \bar{x}_4 \vee \bar{x}_1 \bar{x}_2 \bar{x}_3 \bar{x}_4$   $\Phi$ -Boole-függvényhez tartozó  $\Phi$ -adatgráf.

18. ábra

Ez a  $\Phi$ -adatgráf lehetővé teszi a konjunkciók implikáns voltának az előzőknél egyszerűbb eldöntését. Nevezetesen, implikáns minden olyan konjunkció, amelynek a függvény egyetlen 0-pontja sem pontja.

Miután az algoritmus  $\hat{f}$  bizonyos primimplikánsait írja fel és a  $\Phi$ -adatgráfban nem szerepel  $\hat{f}$  minden pontja, lehetséges, hogy az  $f$  n-nél kisebb szomszédszámú pontjainak egy része nem szerepel a  $\Phi$ -adatgráfban. Ennek következtében az  $n$  szomszéddal rendelkező pontokra is meg kell kísérelni a primimplikáns keresését.

Az alábbiakban ismertetünk egy olyan algoritmust, amely meghatározza va-



lamely  $\Phi$ -Boole-függvény primimplikánsainak IDNF-ák felírását biztosító halmazát és a diszjunkt tartományokat.

### C algoritmus

Jelölje  $E$  az  $f$ ,  $Q$  az  $f_0$  pontjainak halmazát,  $L$  az  $E$  halmaz elemei számát.

1. Rendezzük az  $E$  elemeket szomszédok számának növekvő sorrendje szerint:

$i \leftarrow 0; m \leftarrow L.$

2.  $i \leftarrow i+1$

3. Ha  $P_i$  fel nem dolgozandó, akkor áttérés a 2. lépésre

4. Ha  $s_i = n$ , akkor áttérés a 11. lépésre.

5.  $P_i \in E$ -re  $K_i$  szerint illeszkedő  $p$  konjunkció felírása

6. Ha  $p \rightarrow f$ , akkor áttérés a 11. lépésre.

7.  $p$   $f$ -beli pontjainak a  $p$ -re utaló indexszel és a figyelmen kívül hagyhatóság jelével való ellátása.

$m \leftarrow$  feldolgozandó pontok száma

8. Ha  $p$  nem lényeges primimplikáns, akkor áttérés a 11. lépésre.

9. Ha  $m=0$ , akkor az algoritmus befejeződik

10. Ha  $i=m$ , akkor az algoritmus befejeződik, egyébként a 2. lépéssel folytatódik.

11. Ha  $K_i$ -ből már nem lehet primimplikáns előállítására alkalmas  $C$  koordinátahalmazt kiválasztani, akkor az eljárás a 9. lépéssel folytatódik.

12.  $C$  kiválasztása

$P_i$ -re  $C$  szerint illeszkedő  $p$  konjunkció felírása

Áttérés a 6. lépésre

### II.3.3. Általános Boole-függvények minimalizálása

A II.1.3. pontban érintettük az általános Boole-függvények minimalizálásának kérdését. Ebben a pontban a gráf-módszer olyan általánosítását ismertetjük, amely alkalmas általános Boole-függvények együttes IDNF-inak felírását biztosító együttes primimplikánsok halmazának, valamint a megfelelő



diszjunkt tartományoknak a meghatározására.

Ez a módszer lényegében a II.3.2.-ben leírt módszer alkalmazása az adatgráf olyan további általánosítására, amely az  $F = \{f_1, f_2, \dots, f_M\}$  általános Boole-függvényekre vonatkozó információt is tartalmazza. Ezt a gráfot a továbbiakban általános adatgráfnak nevezzük és a komponens függvényekhez tartozó adatgráfok következő módon képzett egyesítésekként kapjuk. Az általános adatgráf szögpontjainak halmaza két halmazból, az  $F_1$  és  $F_0$  halmazból áll. A  $F_1$  ill.  $F_0$  pontjainak halmaza a komponens  $\Phi$ -Boole-függvényekhez tartozó  $f$  ill.  $f_0$  pontjai halmazainak egyesítései. Minden csucshoz indexként hozzárendeljük azon komponens függvények sorszámait, amelyekhez tartozik. Az egyesítésnél az egy csucsra illeszkedő élek halmaza a komponens függvények  $\Phi$ -adatgráfjaiban az adott csucsra illeszkedő élek halmazainak egyesítésekként áll elő. Az élek közül valódi lesz minden olyan él, amely legalább egy gráfban valódi élként szerepel.

Az alábbiakban megadott példán szemléltetjük az általános adatgráf kialakítását /19. ábra/.

Legyen  $F = \{f_1, f_2, f_3, f_4\}$ , ahol

$$f_1 = x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee x_1 x_2 x_3 \bar{x}_4$$

$$f_2 = x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 \vee x_1 x_2 x_3 \bar{x}_4 \vee x_1 \bar{x}_2 x_3 x_4 \vee \bar{x}_1 \bar{x}_2 \bar{x}_3 x_4$$

$$f_3 = x_1 \bar{x}_2 \bar{x}_4 \vee x_1 x_2 \bar{x}_3 x_4$$

$$f_4 = x_1 \bar{x}_3 \bar{x}_4 \vee x_1 x_2 \bar{x}_4 \vee x_1 x_2 \bar{x}_3 \vee x_1 \bar{x}_2 x_3 x_4$$

$$f_{1,0} = \bar{x}_1 \bar{x}_2 x_3 x_4 \vee x_1 \bar{x}_2 x_3 \bar{x}_4$$

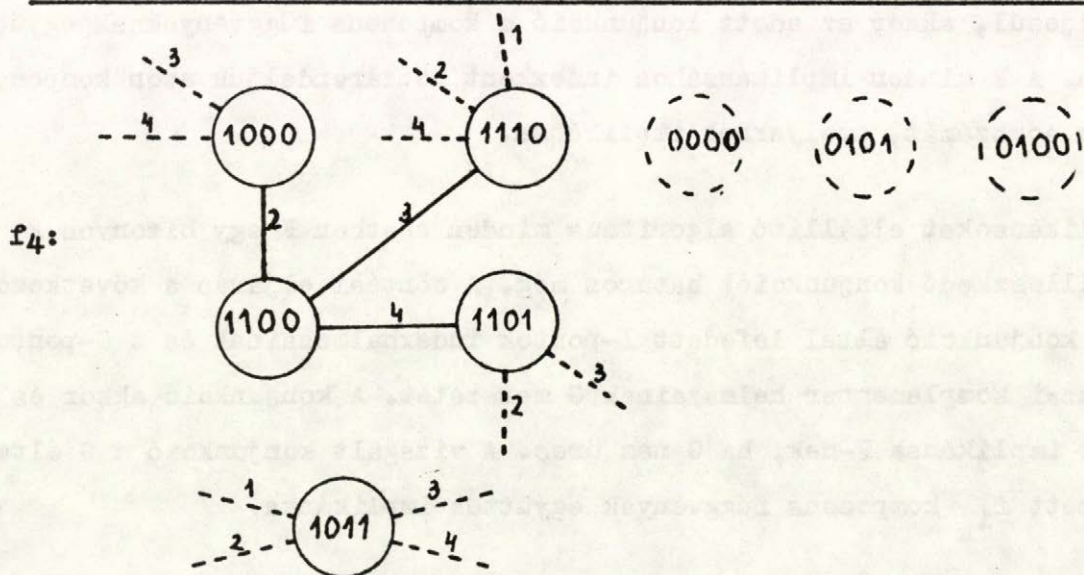
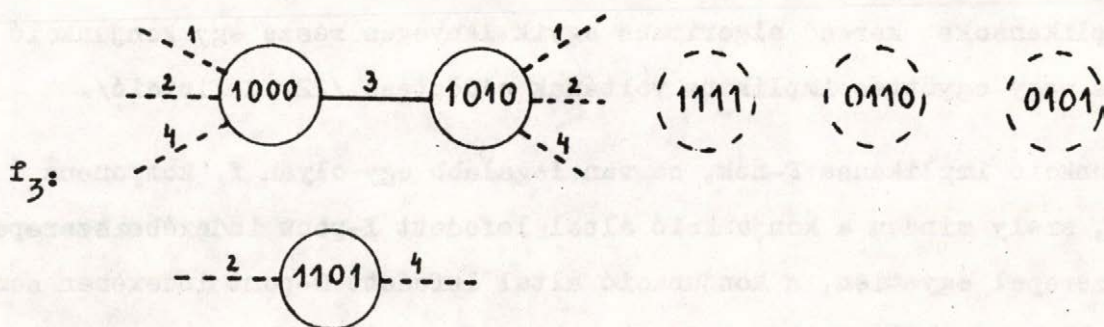
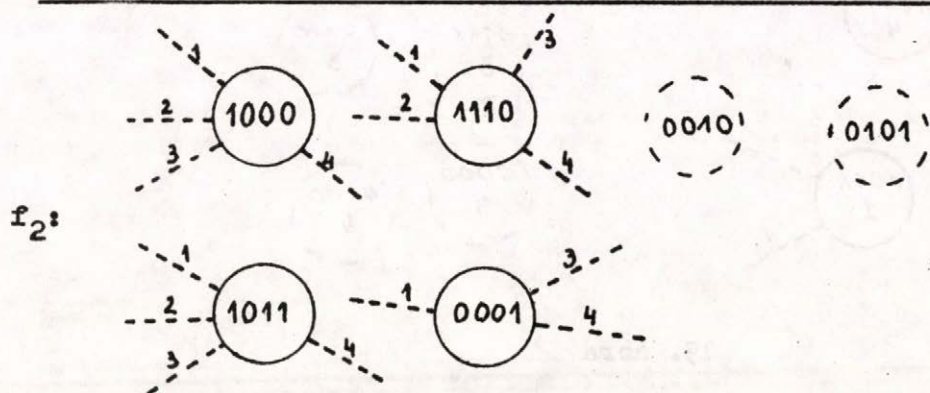
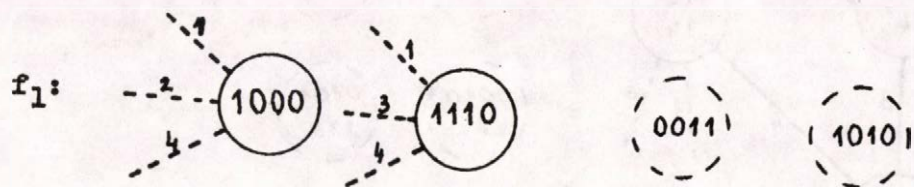
$$f_{2,0} = \bar{x}_1 \bar{x}_2 x_3 \bar{x}_4 \vee \bar{x}_1 x_2 \bar{x}_3 x_4$$

$$f_{3,0} = x_1 x_2 x_3 x_4 \vee \bar{x}_1 x_2 x_3 \bar{x}_4 \vee \bar{x}_1 x_2 \bar{x}_3 x_4$$

$$f_{4,0} = \bar{x}_1 \bar{x}_3 \bar{x}_4 \vee \bar{x}_1 x_2 \bar{x}_3 x_4$$

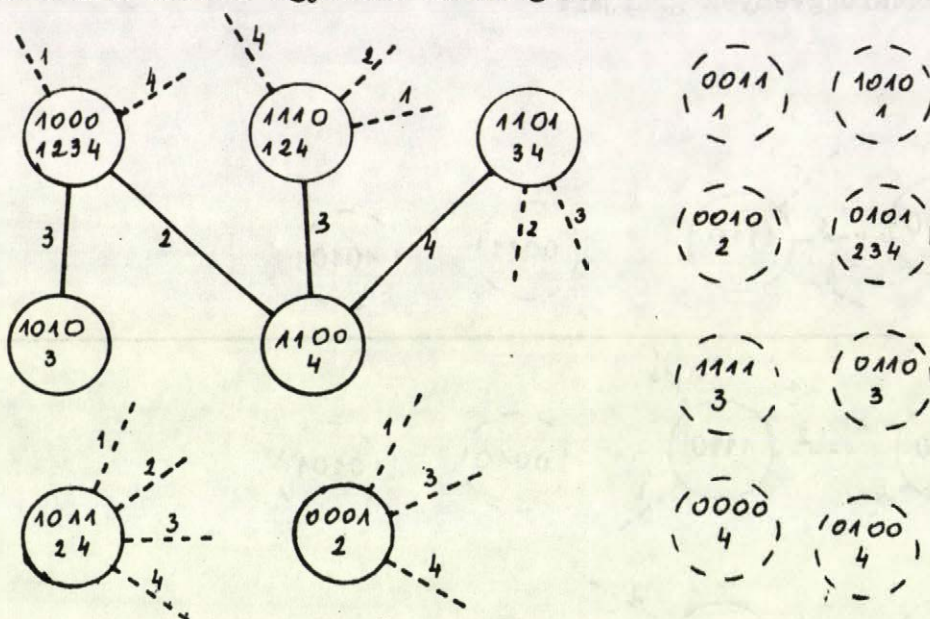


Az egyes koordinátafüggvények gráfjai:





Az F-hez tartozó együttes Boole-gráf:



19. ábra

A primimplikánsokat kereső algoritmus egyik lényeges része egy konjunkció implikáns vagy együttes implikáns voltának eldöntése /22. definíció/.

Egy konjunkció implikánsa F-nek, ha van legalább egy olyan  $f_1$  komponens függvény, amely minden a konjunkció által lefedett 1-pont indexében szerepel, de nem szerepel egyetlen, a konjunkció által lefedett 0-pont indexében sem. Ha valamely konjunkció esetén a fenti feltétel egynél több komponens függvényre teljesül, akkor az adott konjunkció e komponens függvényeknek együttes implikánsa. A F minden implikánsához indexként hozzárendeljük azon komponens függvények sorszámát, amelyeknek implikánsa.

A primimplikánsokat előállító algoritmus minden esetben F egy bizonyos F pontjára illeszkedő konjunkciót határoz meg. A döntési eljárás a következő: vesszük a konjunkció által lefedett 1-pontok indexhalmazainak és a 0-pontok indexhalmazai komplementer halmazainak G metszetét. A konjunkció akkor és csak akkor implikánsa F-nek, ha G nem üres. A vizsgált konjunkció a G által meghatározott  $f_{1_k}$  komponens függvények együttes implikánsa.



Az  $f_{i_1}, f_{i_2}, \dots, f_{i_k}$  együttes primimplikánsának egy olyan konjunkció felel meg, amely implikánsa az  $f_{i_1}, f_{i_2}, \dots, f_{i_k}$  komponens függvényeknek, de a konjunkcióban szereplő változók számát csökkentve ez már nem áll fenn.

A  $F$  függvénynek egy  $P$  pontjára illeszkedő összes együttes primimplikánsa a  $P$  pontra illeszkedő összes olyan konjunkció, amely a  $P$  pont indexei által meghatározott  $f_{i_1}, f_{i_2}, \dots, f_{i_k}$  komponens függvények valamely részhalmazainak együttes primimplikánsa. E primimplikánsok indexhalmazai  $H$  egyesítésének elő kell állítania a  $P$  indexhalmazát, azaz  $\{i_1, i_2, \dots, i_k\} = H$ . Ha van olyan részhalmaza az  $\{i_1, i_2, \dots, i_k\}$ -nak, amely a  $P$  pontra illeszkedő együttes primimplikánsok indexhalmazai közül csak egynek részhalmaza, akkor ezen indexhalmaz által meghatározott komponens függvények együttes primimplikánsa lényeges primimplikánsa  $F$ -nek /24. definíció/.

A feladat értelemszerűen maga után vonja a diszjunkt tartományok fogalmának módosítását. Két pont akkor és csak akkor tartozik egy diszjunkt tartományba, ha a primimplikánsokhoz való tartozást mutató indexeken kívül a komponens függvényekhez való tartozást mutató indexeiknek is megegyeznek.

A  $F = \{f_1, f_2, \dots, f_m\}$  függvény együttes primimplikánsai olyan halmazát határozza meg a következő algoritmus, amely biztosítja a függvény IDNF-inak felírását és a megfelelő diszjunkt halmazok kialakítását.

#### D algoritmus

Jelölje  $E$  a  $F$  függvény 1-pontjainak,  $Q$  a 0 pontjainak halmazát,  $L$  pedig az  $E$  elemeinek számát.

1. Rendezzük az  $E$  elemeket szomszédai számának növekvő sorrendje szerint:

$$i \leftarrow 0; m \leftarrow L$$

2.  $i \leftarrow i + 1$

3. Ha  $P_i$  fel nem dolgozandó, akkor áttérés a 2. lépésre

4. Ha  $s_i = n$ , akkor áttérés a 14. lépésre



5.  $P_1 \in E$ -re  $K_1$  szerint illeszkedő  $p$  konjunkció felírása
6. A  $p$  konjunkció  $G$  indexhalmazának kiszámítása
7. Ha  $G$  üres, akkor áttérés a 14. lépésre
8. A  $p$  primimplikáns azon pontjainak indexhalmazaiból, amelyek  $F$ -nek  $l$ -pontjai, töröljük a  $p$  indexeivel közös indexeket és ellátjuk  $e$  pontokat a  $p$ -re utaló indexszel. A  $F$  azon  $l$ -pontjai, amelyeknek a komponens függvényekhez tartozását mutató indexhalmaza üres, fel nem dolgozandó pontok.
9. Ha  $P_1$ -nek a komponens függvényekhez tartozást mutató indexhalmaza nem üres, akkor áttérés 14. lépésre.
10. Ha  $p$  nem lényeges primimplikáns, akkor áttérés a 14. lépésre.
11.  $m \leftarrow$  feldolgozandó pontok száma
12. Ha  $m=0$ , akkor az algoritmus befejeződik
13. Ha  $i=L$ , akkor az algoritmus befejeződik, egyébként a 2. lépéssel folytatódik
14. Ha  $K$  -ből már nem lehet primimplikáns előállítására alkalmas  $C$  koordinátahalmazt kiválasztani, akkor az eljárás a 11. lépéssel folytatódik.
15.  $C$  kiválasztása  
 $P$  -re  $C$  szerint illeszkedő  $p$  konjunkció felírása  
 Áttérés a 6. lépésre.



### III. Fejezet

#### BOOLE-FÜGGVÉNYEK SZINTÉZISE KÜLÖNBÖZŐ BÁZISOKBAN BIZONYOS

#### KORLÁTOZÁSOK FIGYELEMBEVETELÉVEL

E fejezetben ismertetésre kerülő szintézis módszerek feladata tetszőleges DNF-jukkal adott Boole-függvények olyan nem redundáns zárójeles formulák alakjában történő előállítására  $\{\wedge, \vee, \neg\}$  vagy  $\{\text{NAND}\}$  ill.  $\{\text{NOR}\}$  bázisban, amelyek eleget tesznek bizonyos feltételeknek.

E feltételek egyrészt a formula szerkezetére, másrészt a formulában szereplő bázisfüggvényekre vonatkoznak. A formula szerkezetére vonatkozó feltételek a függvényt realizáló kombinatorikus hálózat szerkezetére vonatkozó technológiai követelmények következményei, amelyek a kimenetek terhelhetőségének és a hálózat mélységének korlátozásából adódnak. A formulában szereplő bázisfüggvényekre vonatkozó feltétel általában a függvények változói számának korlátozását jelenti, amely az e függvényeket realizáló funkcionális elemek bemenetszámaira vonatkozó műszaki követelmények tükröződése.

A berendezések előállításával kapcsolatos műszaki követelmények azonban nemcsak az algoritmusok eredményeként adódó formulákra vonatkozó feltételeket határozzák meg, hanem a feladat megfogalmazásának is irányt szabnak. Ez elsősorban azt jelenti, hogy nem szükséges, sőt gazdaságossági szempontokból nem is célszerű a függvények minimalizálását tűzni ki célul, ezért a feladat általában a Boole-függvények szintézise. E műszaki követelmények korlátozzák továbbá a vizsgálandó függvények körét is, amennyiben kizárják a függvények változói bizonyos értékkombinációinak előfordulását, ami azt jelenti, hogy a realizálandó függvény általában  $\Phi$ -Boole-függvény. A továbbiakban ezért figyelmünket a  $\Phi$ -Boole-függvényekre összpontosítjuk.

Az  $f$   $\Phi$ -Boole-függvény megadható bármely olyan teljesen meghatározott Boole-függvények segítségével, amelyek meghatározzák az  $f$   $\Phi$ -Boole-függvénnyel kompatibilis teljesen meghatározott Boole-függvények halmazát /17.



definíció/.

Ilyen függvényekként választhatók elsősorban  $\hat{f}$  /18. definíció/ és  $\hat{f}$  /19. definíció/, amelyek segítségével  $f$  a következő alakban állítható elő:  
 $f = \hat{f} \vee \gamma \hat{f}$ , ahol  $\gamma$  tetszőleges teljesen meghatározott Boole-függvény.

Az  $f \in \Phi$ -Boole-függvény megadható az  $f_0, f_1 (=f)$  és  $g$  függvények közül bármely kettővel is, ahol  $f_0$  az  $f$  0-pontjai halmazának,  $g$  pedig az  $f$  meghatározatlansági tartományának karakterisztikus függvénye. Szokás az  $f$  megadása  $f_1$  és  $f_0$  ill.  $f_1$  és  $g$  segítségével.

Az  $f$  függvény  $f_0$  és  $f_1$  függvényekkel történő megadása használatos egyes szintézis ill. minimalizáló módszereknél, mint pl. a kipróbálás módszere [27, 48], topológiai módszer [66], gráf-módszer /II.3./ és egyes speciális feladatok megoldásánál /pl. egy függvény DNF-jában biztosan szereplő minimális számú változó megkeresése [37. I/50]. Ez a megadási mód igen gazdaságos a kevés helyen meghatározott függvények esetében. Az  $f \in \Phi$ -Boole-függvény  $f_1$  és  $g$  segítségével  $f = f_1 \vee \gamma g$  alakban állítható elő.

Ez a megadási mód különösen előnyös az olyan számítási eljárásoknál, amelyek során az előálló függvények meghatározatlansági tartományai változnak.

Ezen előállítási módból látható az, hogy egy teljesen meghatározott  $f$  Boole-függvény felfogható olyan  $\Phi$ -Boole-függvényként, amelynek  $f = f_1 \vee \gamma g$  előállításában  $g \equiv 0$ . Ebből következik, hogy a továbbiakban  $\Phi$ -Boole-függvényekre kidolgozott eredményeink speciális esetként vonatkoznak a teljesen meghatározott Boole-függvényekre is, mivel  $g \equiv 0$  esetén  $f = \hat{f} = \hat{f}$ .

A III.1. pontban részletezzük a későbbi pontokban kidolgozott szintézis algoritmusokhoz vezető Akers-féle táblázat módszert és néhány, nem normálformában történő  $\{\wedge, \vee, \neg\}$  bázisbeli, valamint  $\{\text{NOR}\}$  ill.  $\{\text{NAND}\}$  bázisbeli szintézis eljárást.

A III.2. pontban az Akers-féle táblázattal [5] analóg T táblázat tulajdon-



ságait vizsgáljuk. Ennek során bebizonyítunk néhány, a kidolgozásra kerülő szintézis módszer szempontjából hasznos tételt és megfogalmazzuk a  $\Phi$ -Boole-függvények  $\{\wedge, \vee, \neg\}$  bázisbeli zárójeles formulában való szintézisét a T táblázat segítségével.

A III.3. és a III.4. pontokban a disszertációban kidolgozott algoritmusokat ismertetünk a Boole-függvények  $\{\wedge, \vee, \neg\}$  ill.  $\{\text{NAND}\}$  bázisbeli zárójeles formulában való felírására a táblázat módszer alapján, ahol a táblázatot a függvények alsó és felső határaival helyettesítjük. A szintézis során figyelembe vesszünk bizonyos korlátozásokat is.

A III.5. pontban általános Boole-függvények szintézisére is alkalmas  $\{\wedge, \vee, \neg\}$  és  $\{\text{NAND}\}$  bázisbeli szintézisre szolgáló eljárást ismertetünk. Az eljárás a III.3., III.4. pontban leírt algoritmusokban figyelembe vett korlátozásokon kívül figyelembe veszi a funkcionális elemek kimeneteinek terhelhetőségét is.

### III.1. Ismert módszerek

Az egyes szintézis eljárások tanulmányozása során az Akers-féle táblázat módszer vezetett az e fejezetben ismertetésre kerülő algoritmusok kidolgozásához, azért ezzel a módszerrel foglalkozunk részletesebben [5].

Akers módszere közvetlenül az általa logikailag passzívnek nevezett /monoton növekvő/ függvényekre alkalmazható. Természetesen tetszőleges  $\Phi$ -Boole-függvény átalakítható logikailag passzív  $\Phi$ -Boole-függvénnyé, ha a változók benne szereplő negáltjai helyett új változókat vezetünk be. Akers az így átalakított - megnövelt változószámu -  $\Phi$ -Boole-függvényekre dolgozta ki az algoritmust, amely feltételezi, hogy az ilyen  $f$  függvény az  $f_1$  és  $f_0$  függvények igazságtáblájával adott.

Az algoritmus első részében az igazságtáblák méreteinek csökkentése érdekében olyan új  $f_1$  és  $f_0$ -t alakít ki, amelyek az eredeti  $f$   $\Phi$ -Boole-függvény-



nyel kompatibilis  $\Phi$ -Boole-függvényt határoznak meg.

Az algoritmus második részében ezen új igazságtáblák alapján összeállít egy olyan A táblázatot, amelynek sorait ill. oszlopait szegélyezi az  $f_1$  ill.  $f_0$  igazságtáblájából bizonyos szabályok alapján kiolvasható elemi konjunkciókkal; soraiban és oszlopaiban pedig a megfelelő konjunkciókkal kapcsolatos információ helyezkedik el.

Az A táblázat  $A_{ij}$  eleme azon változók halmaza, amelyek közösek az i. sornak és a j. oszlopnak megfelelő elemi konjunkciókban. Cikkében Akers megjegyzi, hogy a függvény logikailag passzív volta miatt az A táblázat egyetlen eleme sem üres. Megállapítja, hogy a sorok elemeiből képzett konjunkciók diszjunkciójával DNF-ban adódó teljesen meghatározott  $f_A$  függvény kompatibilis az  $f$   $\Phi$ -Boole-függvénnyel - az  $f_A$  függvényt a továbbiakban az A táblázat által generált függvénynek nevezzük. Az  $f_A$  függvény KNF-ban is előállítható - az oszlopok elemeiből képzett diszjunkciók konjunkciójaként. A szerző felsorol továbbá olyan szabályokat, amelyeknek az A táblázatra való alkalmazásával előálló  $A'$  táblázat által generált  $f_{A'}$  függvény szintén kompatibilis lesz a kiindulási  $f$  függvénnyel.

E szabályok a következők:

1. A sorok /oszlopok/ felcserélése
2. A "lefedő" sorok ill. oszlopok elhagyása
3. Egy változó egy sorból ill. oszlopból való törlése
4. A táblázat egy-egy eleméből egy kivételével az összes változó törlése.

Ha az A táblázatra a 2., 3., 4. szabályokat mindaddig alkalmazzuk, ameddig lehetséges, akkor az előállt táblázat által generált függvény IDNF-ban ill. IKNF-ban adódik. A minimalizálás érdekében célszerű az 1., 3., 4. szabályokat úgy alkalmazni, hogy minél nagyobb méretű, egyetlen változót tartalmazó résztáblázatok álljanak elő.

Akers a cikkében kimutatja az összefüggést a táblázat által generált és az e táblázat szétvágásával keletkező résztáblázatok által generált függvények



között. Nevezetesen, ha  $A_1$  és  $A_2$  az  $A$  táblázat sorok szerinti szétvágásával keletkezik, akkor  $f_A = f_{A_1} \vee f_{A_2}$ , ha pedig oszlop szerinti szétvágásával, akkor  $f_A = f_{A_1} \wedge f_{A_2}$ .

Az  $A_1$  és  $A_2$  táblázat összeillesztésével keletkező  $A$  táblázat által generált függvény és a résztáblázataik által generált függvények között természetesen ismét ugyanilyen összefüggések állnak fenn. Pl. a 20. ábrán szemléltetett  $A$  táblázat esetén:

$$f_A = (f_{A_1} \vee f_{A_2}) \wedge (f_{A_3} \vee (f_{A_4} \wedge f_{A_5}))$$

$A_1$	$A_3$	
$A_2$	$A_4$	$A_5$

20. ábra

A szerző megmutatja azt is, hogy hogyan adható meg az  $A$  táblázat alkalmas szétvágásával az  $f_A$  függvény valamely dekompozíciója. Ha az  $A$  táblázat szétvágása olyan, hogy az  $A_1$  résztáblázatok által generált függvényekre  $f_{A_1}$  valamelyik változóval egyenlő, akkor végeredményben az  $f_A$  függvény egy, általában többmélységű,  $\{\wedge, \vee, \neg\}$  bázisban felírt zárójeles formula alakjában áll elő.

Akers megemlíti, hogy a szintézis során figyelembe vehetők különböző korlátozások is, de nem foglalkozik ezekkel a kérdésekkel. Módszere csak az  $\wedge, \vee$  műveletek segítségével /monoton/ felírható függvényekre alkalmazható közvetlenül; az  $\{\wedge, \vee, \neg\}$  bázistól különböző bázisbeli problémát pedig nem is érinti.

A módszer további hiányossága, hogy a szerző az általa vizsgált speciális esetekben is – indokolatlanul – a RDNF-ák alapján írja fel az  $A$  táblázatot, így látszólag feltételezi, hogy azt előzőleg előállítottuk.



Az A táblázat részletes vizsgálatával olyan eredményekhez jutottunk, amelyek alkalmasak az Akers-módszer hiányosságainak kiküszöbölésére és az e fejezet bevezetésében említett irányokba való továbbfejlesztésére. Eppen ezért az Akers-féle táblázat módszertől eltérő eljárásokkal nem foglalkozunk részletesen.

Az  $\{\wedge, \vee, \neg\}$  bázisbeli zárójeles formulát eredményező szintézis eljárások egyik csoportja tekinthető a Quine-McCluskey módszer továbbfejlesztésének is. Az [1 és 41]-ben leírt módszerekkel minimális számú változót tartalmazó zárójeles formulát érnek el; a maximális zárójelmélység kettő. Alapgondolatuk, hogy a konjunkción kívül a konjunkciók diszjunkcióját is primér kifejezésnek tekintik. E primér kifejezések közül kiválasztják a függvényt lefedő minimális készletet. A primér kifejezések osztályozása és a minimális készlet kiválasztása mindkét cikkben lényegében bizonyos korlátozások melletti próbálgatás.

Röviden érintjük Bereczky Ilona eddig még nem publikált munkáját, amely a fenti szintézis feladat egy újszerű megközelítése. Az általa kidolgozott rekurzív eljárás minimális zárójeles formulát eredményez  $\{\wedge, \vee, \neg\}$  bázisban. Az eljárás érdekessége, hogy az egyes Boole-függvények optimális szintézise helyett a Boole-függvények Pólya-féle osztályai alapján előállított függvényosztályok reprezentánsai optimális szintézisét végzi el.

Az  $\{\wedge, \vee, \neg\}$  bázisbeli zárójeles szintézis probléma mellett egyre többen foglalkoznak  $\{\text{NAND}\}$  ill.  $\{\text{NOR}\}$  bázisbeli szintézissel és szinte kivétel nélkül hangsúlyozzák a tetszőleges bázisban való szintézis fontosságát. Az eljárások egy részében, amelyre példaként [31]-et említjük, lényeges szerepet játszanak a formulákkal végzett bizonyos műveletek. Más, pl. a [40, 70]-ben leírt eljárások elsősorban formula átalakítási azonosságokat használnak fel; ide sorolható a [43] cikk is. Az utóbbi időben egyre többen foglalkoznak Boole-algebrában első látásra formális műveletek definiálásával és vizsgálatával [13, 14, 24, 57], amelyek azonban egyes esetekben pl. a szintézisre vonatkozóan is eredményeket adnak [44, 58, 64, 65]. Ugy tűnik, hogy az



ilyen irányú kutatások eredményeként a szintézis-eljárás és az automaták logikai tervezésének bizonyos fázisai, valamint a szintézis eredményeinek realizálás utáni ellenőrzése /diagnosztika/ egységesen lesz kezelhető.

### III.2. A T táblázat

Részletesen elemeztük az Akers-féle táblázat módszernél a kiindulási  $\Phi$ -Boole-függvény és az algoritmus első részének eredményeként kapott igazságtábla alapján megkonstruált A táblázat szegélyezése közötti közvetlen kapcsolatot. A vizsgálat eredményeként megállapítottuk, hogy az A táblázat szegélyezésében szereplő konjunkciók az eredeti  $\Phi$ -Boole-függvénnyel kompatibilis  $f \Phi$ -Boole-függvény  $\hat{f}$  ill.  $\hat{f}^*$  RDNF-ját állítják elő.

A továbbiakban be fogjuk látni, hogy közvetlenül az eredeti  $\Phi$ -Boole-függvény tetszőleges DNF-ja alapján megkonstruálva egy, az Akers-féle A táblázattal analóg T táblázatot, a táblázatra vonatkozóan az [5]-ben felsorolt tulajdonságok érvényben maradnak. Ezen túlmenően megmutatunk néhány, a táblázatra és a táblázat által generált függvényre vonatkozó tulajdonságot és összefüggést.

Legyenek adva  $q_1, q_2, \dots, q_l$ , valamint  $p_1, p_2, \dots, p_k$  elemi konjunkciók. A továbbiakban az ezek által meghatározott T táblázat alatt egy olyan  $l \times k$ -s táblázatot értünk, amelynek  $A_{ij}$  eleme a  $q_i$  és  $p_j$  konjunkciók mindegyikében azonos kitévővel /vagy mindkettőben negátlanul, vagy mindkettőben negálva/ szereplő közös változók halmaza. Szükség esetén a T táblázat sorait a  $q_1, q_2, \dots, q_l$ , oszlopait pedig a  $p_1, p_2, \dots, p_k$  elemi konjunkciókkal szegélyezzük /21. ábra/.

Legyenek az  $f \Phi$ -Boole-függvényre  $\hat{f}^* = \bigvee_{i=1}^l q_i$  és  $\hat{f} = \bigvee_{i=1}^k p_i$ ; az  $f \Phi$ -Boole-függvényhez tartozó táblázatnak nevezzük a  $q_1, q_2, \dots, q_l$  és a  $p_1, p_2, \dots, p_k$  által meghatározott T táblázatot.



	$p_1$	$p_2 \dots p_j \dots p_k$
$q_1$		$\vdots$
$q_2$		$\vdots$
$\vdots$		$\vdots$
$q_i$	$\dots \dots \dots$	$A_{ij} \dots \dots$
$\vdots$		$\vdots$
$q_\ell$		$\vdots$

21. ábra

Könnyű belátni, hogy  $f^* = \hat{f}^* \vee \gamma_{\hat{f}} f^*$  miatt az  $f$   $\Phi$ -Boole-függvényhez tartozó T táblázat az  $f$ -hez tartozó T táblázatnak transzponáltja /amely a T táblázatból sorainak oszlopaival történő felcserélésével áll elő/.

Ugyanígy látható, hogy az  $\bar{f} = \overline{\hat{f}} \vee \gamma_{\hat{f}} \bar{f}$  függvényhez tartozó T táblázat az  $f$ -hez tartozó T táblázat transzponáltjából a táblázatban szereplő változók kitevőinek megváltoztatásával áll elő.

Megjegyzés: Mivel  $f_0 = \hat{f}$  és ha  $\bar{f} = \bigvee_{i=1}^m v_i$ , akkor  $f^* = \bigvee_{i=1}^m v_i'$ , ahol a  $v_i$  és  $v_i'$  elemi konjunkciók csak a bennük szereplő változók kitevőjében különböznek; a T táblázat előállítható  $f_0$  és  $f_1$  DNF-ja alapján is.

Illusztrációként megadjuk az  $\hat{f} = ab \vee ac\bar{d} \vee bc \vee \gamma \bar{b}d$  függvény által generált T táblázatot /22. ábra/.

$$\hat{f} = ab \vee ac\bar{d} \vee bc, \quad \hat{f}^* = a\bar{b}c \vee acd \vee bcd \vee abd$$

$\hat{f}^* \backslash \hat{f}$	ab	acd	bc
$\bar{a}\bar{b}c$	a	a,c	c
acd	a	a,c	c
bcd	b	c	b,c
abd	a,b	a	b

22. ábra

Logikailag passzív függvények A táblázatára Akers megállapítja, mi pedig



tetszőleges  $\Phi$ -Boole-függvény T táblázatára bebizonyítjuk az alábbi tételt.

### Tétel:

Az  $f \in \Phi$ -Boole-függvényhez tartozó T táblázatban egyetlen  $A_{ij}$  sem üres.

### Bizonyítás:

Tegyük fel, hogy a T táblázatnak van üres eleme, legyen ez az  $A_{ij}$ . Legyen az  $i$ -edik sort szegélyező elemi konjunkció  $\vee$ , a  $j$ -edik oszlopot szegélyező elemi konjunkció  $\mu$ . Ez azt jelenti, hogy a  $\vee$  implikánsa  $\hat{f}^*$ -nak, a  $\mu$  implikánsa az  $\hat{f}$ -nek. Egy függvény duálisának és negáltjának implikánsai közötti kapcsolat alapján  $\vee'$  implikánsa  $\hat{\hat{f}}$ -nak. Ha  $A_{ij}$  üres, akkor  $\mu \vee' \neq 0$ , tehát  $f_0$  és  $f_1$ -nek van közös implikánsa, ami lehetetlen.

Valamely  $f \in \Phi$ -Boole-függvény  $\hat{f}$  és  $\hat{f}^*$  adott DNF-i alapján az  $f$ -hez tartozó T táblázat egyértelműen meghatározott, azonban különböző DNF-ák különböző T táblázatokat állíthatnak elő. Nyilvánvaló, hogy egy adott T táblázat több  $\Phi$ -Boole-függvényhez is tartozhat.

Egy adott T táblázat által generált függvénynek nevezzük a  $g = \bigvee_{j=1}^k \prod_{i=1}^l a_{ij}$ ,  $h = \bigvee_{i=1}^l \prod_{j=1}^k a_{ij}$  függvények által meghatározott  $f_T = g \vee h^*$  függvényt, ahol  $a_{ij}$  az  $A_{ij}$ -ben szereplő változók konjunkciója és  $g$  tetszőleges teljesen meghatározott Boole-függvény. Az  $f_T$  függvényre érvényes az alábbi tétel.

### Tétel:

A T által generált  $f_T$  függvény

a./  $\Phi$ -Boole-függvény

b./ kompatibilis bármely olyan  $f \in \Phi$ -Boole-függvénnyel, amelyhez a T táblázat tartozik, vagyis  $f \leq f_T \leq \hat{f}$ .

### Bizonyítás:

a./ Elegendő belátni, hogy  $g \leq h^*$ . A duális képzési szabálya alapján

$$h^* = \bigvee_{j=1}^k \prod_{i=1}^l a_{ij}^*. \text{ Mivel egy } \sigma \text{ elemi konjunkció mindig implikálja a duálisát, } \sigma^* \text{-ot, ezért minden } i, j \text{-re } a_{ij}^* \geq a_{ij}, \text{ tehát}$$

$$h^* = \bigvee_{j=1}^k \prod_{i=1}^l a_{ij}^* \geq \bigvee_{j=1}^k \prod_{i=1}^l a_{ij} = g.$$



b./ Az a./ alapján elegendő megmutatni, hogy  $f \leq g$  és  $h^* \leq \hat{f}$ . A T táblázatot szegélyező  $q_1, q_2, \dots, q_\ell$  és  $p_1, p_2, \dots, p_k$  elemi konjunkciókra teljesül, hogy  $q_i \leq \prod_{j=1}^k a_{ij}$  ( $i=1, 2, \dots, \ell$ ) és  $p_j \leq \prod_{i=1}^{\ell} a_{ij}$ , ( $j=1, 2, \dots, k$ ), mivel az  $a_{ij}$  legfeljebb a  $q_i$  és  $p_j$ -ben szereplő változókat tartalmazza. Ebből következik, hogy a definícióban szereplő  $g$  és  $h$  függvényekre  $f = \bigwedge_{j=1}^k p_j \leq g$  és  $h \geq f^*$ . Az utóbbiból a dualitás miatt  $h^* \leq \hat{f}$ . Ezzel a tételt bebizonyítottuk.

A bebizonyított tétel a./ pontja alapján bevezethetjük a  $g = f_{\wedge T}$  és a  $h^* = \hat{f}_T$  jelölést.

A következő tétel biztosítja, hogy a T táblázat felírásához az  $f$  és  $\hat{f}^*$  tetszőleges DNF-i felhasználhatók.

#### Tétel:

A T táblázat által generált  $f_T$  függvény nem függ az  $f$  és  $\hat{f}^*$  megadási módjától.

#### Bizonyítás:

Mivel egy függvény tetszőleges DNF-jában szereplő konjunkciók megkaphatók a függvény KDNF-jából az összevonási /1/ szabály alkalmazásával, elegendő belátni, hogy alkalmazva ezt a szabályt  $f$  vagy  $\hat{f}^*$  valamely alakjára és felírva a T táblázatot az eredményül kapott DNF alapján,  $f_{\wedge T}$  és  $\hat{f}_T^*$  nem változik.

Tegyük fel, hogy  $p_t = ax$ ,  $p_s = a\bar{x}$ . Az összevonási szabály alkalmazása azt jelenti, hogy az  $a = p_{k+1}$  konjunkció  $f$  diszjunkciós tagjai közé kerül. A kapott T táblázat bővül az ennek megfelelő oszloppal. Világos, hogy  $f_{\wedge T} = f_{\wedge T}$ . Az  $\hat{f}_T^* = \hat{f}_T^*$  pedig azért áll fenn, mert a T táblázat egyetlen sora sem tartalmaz olyan változókat, amelyeket a T táblázat nem tartalmaz.

Ezzel a tételt bebizonyítottuk.

Vizsgáljuk meg, hogy milyen esetekben lesz biztosan különböző  $f$  és  $f_T$ .

Egyszerűség kedvéért tegyük fel, hogy a T táblázatot  $f$  és  $\hat{f}^*$  RDNF-ja alapján írtuk fel. Mivel egy függvény RDNF-jában azok és csak azok a változók sze-



repelnek, amelyektől a függvény függ, a T táblázatban pontosan azok a változók fordulnak elő, amelyektől mind az  $\underline{f}$ , mind az  $\hat{f}^*$  /azaz  $\hat{f}$ / függ. Ebből következik, hogy az  $\underline{f}_T$  és az  $\hat{f}_T^*$  csak a táblázatban szereplő változók függvénye. Könnyen belátható ezek után az is, hogyha  $\underline{f}$  monoton az  $x_1$  változóban /tehát RDNF-jában az  $x_1$  szerepel, de  $\bar{x}_1$  nem/,  $\hat{f}$  pedig nem monoton  $x_1$ -ben, de függ tőle, akkor  $\underline{f}_T$  és  $\hat{f}_T$   $x_1$ -ben monoton függvények lesznek.

Ezen utóbbi állítás igaz az  $\underline{f}$  és  $\hat{f}$  függvényekre vonatkozó feltételek felcserélésével is. Nem meglepő tehát a következő tétel, amelynek részletes bizonyítására nem térünk ki.

#### Tétel:

Az  $\underline{f} \neq \underline{f}_T$  egyenlőtlenség szükséges és elégséges feltétele az, hogy létezzen legalább egy olyan  $x_1$  változó, amelyre  $\underline{f}$  nem monoton  $x_1$ -ben, de  $\hat{f}$  monoton  $x_1$ -ben, vagy  $\underline{f}$  függ  $x_1$ -től, de  $\hat{f}$  független  $x_1$ -től. Felcserélve  $\underline{f}$  és  $\hat{f}$ -et a feltételekben, az  $\hat{f} \neq \hat{f}_T$  szükséges és elégséges feltételeit kapjuk.

A továbbiakban a T táblázat  $T_1$  résztáblázatai által generált  $\underline{f}_{T_1}$  függvények és az  $\underline{f}_T$  közötti összefüggéseket vizsgáljuk. Jelölje I a T táblázat sorai, J oszlopai sorszámainak halmazát. Legyen  $M \subseteq I$  és  $N \subseteq J$ . A  $T_{MN}$  táblázat a T táblázat azon résztáblázatát jelöli, amelynek elemei az  $i \in M$ ,  $j \in N$  indexű  $A_{ij}$ -k. A  $T_{MN}$  táblázat által generált függvény:

$$\underline{f}_{T_{MN}} = \bigvee_{j \in N} \bigwedge_{i \in M} a_{ij} \vee \gamma \left[ \bigvee_{i \in M} \bigwedge_{j \in N} a_{ij} \right]^*$$

A T táblázat résztáblázatai által generált függvényekre érvényes az alábbi tétel.

#### Tétel:

a./  $\underline{f}_T = \underline{f}_{T_{I,J_1}} \vee \underline{f}_{T_{I,J_2}}$  tetszőleges  $J_1 \cup J_2 = J$  mellett

b./  $\underline{f}_T = \underline{f}_{T_{I_1,J}} \wedge \underline{f}_{T_{I_2,J}}$  tetszőleges  $I_1 \cup I_2 = I$  esetén.



Bizonyítás:

a./  $f_T = f_T \vee \gamma \hat{f}_T$ ;

$$\begin{aligned} f_{T_{I,J_1}} \vee f_{T_{I,J_2}} &= \left( \bigvee_{j \in J_1} \prod_{i \in I} a_{ij} \right) \vee \left( \bigvee_{j \in J_2} \prod_{i \in I} a_{ij} \right) \vee \\ &\vee \gamma \left[ \left( \bigvee_{i \in I} \prod_{j \in J_1} a_{ij} \right)^* \vee \left( \bigvee_{i \in I} \prod_{j \in J_2} a_{ij} \right)^* \right] = \\ &\bigvee_{j \in J_1 \cup J_2} \prod_{i \in I} a_{ij} \vee \gamma \left( \bigvee_{j \in J_1 \cup J_2} \prod_{i \in I} a_{ij}^* \right) = f_T \vee \gamma \hat{f}_T \end{aligned}$$

b./ A tétel a./ pontja és az  $f_T^* = f_T^*$  egyenlőség felhasználásával:

$$f_{T_{I_1 J}} \wedge f_{T_{I_2 J}} = ((f_{T_{I_1 J}} \wedge f_{T_{I_2 J}})^*)^* = (f_{T_{I_1 J}}^* \vee f_{T_{I_2 J}}^*)^* = (f_T^*)^* = ((f_T)^*)^* = f_T$$

A  $T$  táblázattal kompatibilisnek nevezünk egy  $F_T$  teljesen meghatározott Boole-függvényt, ha kompatibilis az  $f_T$ -vel. Nyilvánvaló, hogyha a  $T$  táblázatot egy teljesen meghatározott  $f$  Boole-függvény alapján írjuk fel, akkor  $f = f_T = F_T$ .

Világos, hogy az előző tétel érvényes akkor is, ha az egyes résztáblázatok által generált függvények helyett a résztáblázatokkal kompatibilis valamely teljesen meghatározott függvényeket tekintünk.

A  $T$  táblázattal kompatibilis függvényekre bebizonyítjuk az alábbi tételt.

Tétel:

Ha egy  $T$  táblázatban szereplő  $A_{ij}$ -kre

$\bigcap_{1 \leq i \leq l} A_{ij} = \{x_{m_1}, x_{m_2}, \dots, x_{m_r}\} \neq \emptyset$ , akkor bármely  $x_{m_n}$  ( $n=1, 2, \dots, r$ ) a  $T$ -vel kompatibilis függvény.



### Bizonyítás:

Legyen  $x_{mn} \in \bigcap_{\substack{1 \leq i \leq l \\ 1 \leq j \leq k}} A_{ij}$ . Ekkor  $x_{mn}$  szerepel  $f_T$  és  $\hat{f}_T^*$  minden diszjunkciós

tagjában, ezért  $x_{mn} \geq f_T$  és  $x_{mn} \geq \hat{f}_T^*$ .

A duálisra vonatkozó szabály miatt  $x_{mn} = x_{mn}^* \leq \hat{f}_T$ . Ezzel a tételt bebizonyítottuk.

E tételből a szintézisre a következő egyszerű eljárás adódik. A kiindulási  $T$  táblázatot olyan  $T_{MN}$  résztáblázatokra bontjuk, amelyekre

$\bigcap_{\substack{i \in M \\ j \in N}} A_{ij} = \{x_{m_1}, x_{m_2}, \dots, x_{m_r}\} \neq \emptyset$ . E résztáblázatok mindegyike kompatibilis

egy-egy  $F_{T_{MN}} = x_{mn} \in \bigcap_{\substack{i \in M \\ j \in N}} A_{ij}$  függvénnyel;  $F_T$  a résztáblázatok által gene-

rál függvényekre vonatkozó tételek alapján  $\{\wedge, \vee, \neg\}$  bázisbeli zárójeles formulával áll elő.

### III.3. $\Phi$ -Boole-függvények szintézise $\{\wedge, \vee, \neg\}$ bázisban a határok felhasználásával

Az előzőekben megfogalmaztuk a táblázat-módszerrel való szintézis lényegét. Ebben a pontban egy olyan algoritmust ismertetünk, mely az  $F_T$ -nek a zárójelmélységre és a bázisfüggvények változói számára vonatkozó megkötéseket is figyelembe vevő zárójeles formuláját állítja elő. Ezen algoritmusnak az előző pont eredményei alapján olyan gépi reprezentációját dolgoztuk ki, amely nem a táblázattal, hanem az  $f_T$  és  $\hat{f}_T^*$  függvényekkel /a határokkal/ dolgozik; a táblázat résztáblázatokra való szétvágásának az  $f_T$  és  $\hat{f}_T^*$  megfelelő felbontása felel meg. Ez a gépi reprezentáció a memóriaigény csökkenéséhez vezet.

A továbbiakban megfogalmazzuk azt, hogy a bázisfüggvények változói számára



és a zárójelmélységre vonatkozó megkötések hogyan tükröződnek az algoritmusban. Az, hogy a  $\vee$  ill. az  $\wedge$  bázisfüggvény változóinak maximális száma  $m$ , azt jelenti, hogy egy résztáblázatot maximum  $m$  olyan résztáblázatra bontunk, amely csak oszlopirányú ill. csak sorirányú felbontással keletkezik. A zárójelmélységre tett korlátozás azt jelenti, hogy a táblázat felváltva következő sor- ill. oszlopirányú felbontásainál az oszlopirányú felbontások száma nem lehet e korlátnál nagyobb. Az algoritmus leírásánál használjuk a következő elnevezéseket.

Egy  $f$  függvény adott DNF-jában előforduló negált vagy negálatlan változó súlyának nevezzük azt a számot, ahányszor az illető változó előfordul az adott DNF-ban. Egy változó súlya az  $f_1, f_2$  függvénytárban az egyes függvénybeli súlyaik szorzata.

A leírás egyszerűsítése érdekében egy  $f_T, \hat{f}_T^*$  függvénypárt  $H$ -val jelölünk. Az algoritmus azon lépései során, amelyek valamely  $T_1$  résztáblázat szétvágásának felelnek meg, a következő lépésben feldolgozásra kiválasztott függvénypárt  $H_{i+1}$ -el, a fennmaradót  $H_i$ -vel jelöljük.

A  $H_i$ -hez rendeljük a  $k, \ell, \ell_z$  számokat -  $H_i(k, \ell, \ell_z)$ , ahol  $k$  azt mutatja, hogy  $H_i$  hányadszor jelöl egy táblázat azonos irányú szétvágásának megfelelő felbontással előálló függvénypárt;  $\ell$  azt, hogy  $H_i$  oszlop ( $\ell=1$ ) vagy sorirányú ( $\ell=2$ ) szétvágásnak megfelelő felbontással állt elő,  $\ell_z$ -ben pedig azt számoljuk, hogy a  $H_i$  feldolgozásának befejezésekor hány zárójelet kell az  $F_T$  előálló  $F$  zárójeles formulájában elhelyeznünk.

Jelölje  $m_z$  a megengedett zárójelmélységet,  $m(1)$  ill.  $m(2)$  a  $\vee$  ill. az  $\wedge$  függvény változóinak maximális számát.  $F \Leftarrow \times$  azt jelenti, hogy az  $F$ -ben a soronkövetkező helyre a  $\times$  jel kerül.



## Algoritmus

1.  $i \Leftarrow 1$ ;  $a \Leftarrow 0$ ;  $z \Leftarrow 0$ ;  $F \Leftarrow 0$ ;  $m_i \Leftarrow 0$ ;  
 $H_1(k, l, l_z) \Leftarrow f_{T_1}, \hat{f}_{T_1}^*$ ;  $k \Leftarrow 1$ ;  $l \Leftarrow 1$ ;  $l_x \Leftarrow 1$ .
2. Ha  $m_z = 0$ , akkor az algoritmus a 25. lépéssel folytatódik.
3. Kiválasztjuk a  $H_1$ -ben a maximális súlyu  $x_j$ -t.
4. Ha a  $H_1$  által meghatározott  $\Phi$ -Boole-függvény kompatibilis a max. súlyu  $x_j$ -vel, akkor a 16. lépés következik.
5. Ha  $(l_a = 0) \equiv (l = 1)$ , akkor a 13. lépés következik.
6. Ha  $x = 1$ , akkor  $k \Leftarrow 2$  és a 8. lépés következik.
7.  $k \Leftarrow k + 1$
8.  $l \Leftarrow 2$ ;  $H_1$  sorirányu felbontása  $H_1, H_{1+1}$ -re;  $i \Leftarrow i + 1$ ;  $l \Leftarrow 2$ .
9.  $k \Leftarrow k - 1$ ;  $l_x \Leftarrow l$ ;  $l_z \Leftarrow 0$ .
10. Ha  $x_j$  kompatibilis a  $H_1$  által meghatározott  $\Phi$ -Boole-függvénnyel, akkor a 16. lépés következik.
11. Ha  $z = m_z$ , akkor a 25. lépés következik.
12.  $l \Leftarrow l - 1$  és a 3. lépés következik.
13. Ha  $x = 1$ , akkor  $k \Leftarrow k + 1$  és a 15. lépés következik.
14.  $k \Leftarrow 2$ ;  $l_z \Leftarrow l_z + 1$ ;  $F \Leftarrow ($ ;  $z \Leftarrow z + 1$ .
15.  $l \Leftarrow 1$ ;  $H_1$  oszlopirányu felbontása  $H_1, H_{1+1}$ -re;  
 $i \Leftarrow i + 1$ ;  $l \Leftarrow 1$ ; és a 8. lépés következik.
16.  $F \Leftarrow x_j$
17. Ha  $z \neq 0$ , akkor  $F \Leftarrow ( \dots )^{l_z}$ ;  $z \Leftarrow z - l_z$ .
18. Ha  $i = 1$ , akkor a 28. lépés következik.
19.  $i \Leftarrow i - 1$
20. Ha  $m_i \neq 0$ , akkor  $m_i \Leftarrow m_i - 1$  és a 22. lépés következik.
21. Ha  $k \neq m(l)$ , akkor  $l_a \Leftarrow 0$  és a 23. lépés következik.
22.  $l_a \Leftarrow l$
23. Ha  $l = 1$ , akkor  $F \Leftarrow \vee$  és a 3. lépés következik.
24.  $F \Leftarrow \wedge$  és a 3. lépés következik.
25.  $l_a \Leftarrow 2$
26. Ha  $f_{T_1}$  konjunkciónak  $s$  számára  $s > m(1)$ , akkor hibajelzés, megállás.



27.  $H_i$  oszlopírányu felbontása  $H_i, H_{i+1}, \dots, H_{i+s-1}$ -re  $f_{T_1}$  egyes konjunkcióinak megfelelően;  $i \Leftarrow i+s-1$ ;  $m_q \Leftarrow s$  és a 3. lépés következik.
28. Ha  $z \neq 0$ , akkor hibajelzés, megállás.
29. Az algoritmus befejeződik.

#### III.4. Szintézis {NAND} vagy {NOR} bázisban

A NAND művelet és az  $\wedge, \vee, \neg$  műveletek közötti  $\text{NAND}(f_1, f_2, \dots, f_k) = \overline{f_1 \wedge f_2 \wedge \dots \wedge f_k} = \overline{f_1} \vee \overline{f_2} \vee \dots \vee \overline{f_k}$  összefüggés alapján világos, hogy ha  $\overline{f} = \bigvee_{i=1}^k f_i$ , akkor  $\text{NAND}(f_1, f_2, \dots, f_k) = f$ . Ennek alapján egy lehetséges {NAND} bázisbeli szintézis eljárás a következő.

1.  $\overline{f}$  előállítása  $\overline{f} = \bigvee_{i=1}^k f_i$  alakban.
2. Ha minden  $i$ -re  $f_i$  kompatibilis egy változóval, akkor az eljárás befejeződik. Egyébként az eljárás a változókkal nem kompatibilis  $f_i$  függvényekre az 1. lépéssel folytatódik.

Ez a szintézis eljárás egyszerűen megvalósítható pl. a III.1.-ben ismertetett táblázat-módszerrel. Ugyanis az  $\overline{f}$ -hoz tartozó táblázat /amely az  $f$ -hez tartozó T táblázat negáltja  $\neg T$  lásd III.2./ sorirányu felbontásával kapott  $T_1, T_2, \dots, T_k$  résztáblázatokkal kompatibilis  $f_{T_1}, f_{T_2}, \dots, f_{T_k}$  függvényekre  $f = \text{NAND}(f_{T_1}, f_{T_2}, \dots, f_{T_k})$ . Így az eljárás a következő lehet:

1. Képezzük  $\neg T$ -t
2. Bontsuk fel  $\neg T$ -t  $T_1, T_2, \dots, T_k$  sorirányu résztáblázatokra.
3. Azon  $T_i$  résztáblázatok, amelyekhez van velük kompatibilis változó, tovább nem bontandók. Egyébként a  $T_j$ -kre megismételjük az eljárást az 1. lépéstől mindaddig, amíg minden résztáblázathoz nem találunk vele kompatibilis változót.
4. A résztáblázatokkal kompatibilis változók segítségével az  $F_T$  felírása.



A módszert egy egyszerű példán illusztráljuk.

$$T = \begin{array}{|c|c|c|} \hline a,b & b & a \\ \hline a & c & a,c \\ \hline \end{array}$$

1. lépés

$$\neg T = \begin{array}{|c|c|} \hline \bar{a} & \bar{a}, \bar{b} \\ \hline \bar{c} & \bar{b} \\ \hline \bar{a}, \bar{c} & \bar{a} \\ \hline \end{array}$$

2. lépés. A sorirányu felbontás eredménye:

$$T_1 = \begin{array}{|c|c|} \hline \bar{a} & \bar{a}, \bar{b} \\ \hline \bar{a}, \bar{c} & \bar{a} \\ \hline \end{array},$$

$$T_2 = \begin{array}{|c|c|} \hline \bar{c} & \bar{b} \\ \hline \end{array}$$

3. lépés.  $F_{T_1} = \bar{a}$ , tehát nem foglalkozunk vele.  $T_2$ -re megismételjük az eljárást, tehát:

1. lépés

$$\neg T_2 = \begin{array}{|c|} \hline c \\ \hline b \\ \hline \end{array}$$

2. lépés. A sorirányu felbontás eredménye:

$$T_3 = \begin{array}{|c|} \hline c \\ \hline \end{array} \text{ és } T_4 = \begin{array}{|c|} \hline b \\ \hline \end{array}$$

3. lépés.  $F_{T_3} = c$ ,  $F_{T_4} = b$ . Nincs további felbontásra szükség.

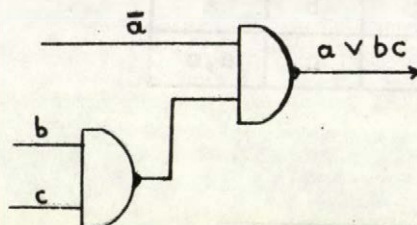
4. lépés.

$F_T = \text{NAND}(F_{T_1}, F_{T_2})$ ,  $F_{T_2} = \text{NAND}(F_{T_3}, F_{T_4})$  így

$F_T = \text{NAND}(F_{T_1}, \text{NAND}(F_{T_3}, F_{T_4})) = \text{NAND}(a, \text{NAND}(c, b))$ .

A függvényt realizáló hálózat a 23. ábrán látható.





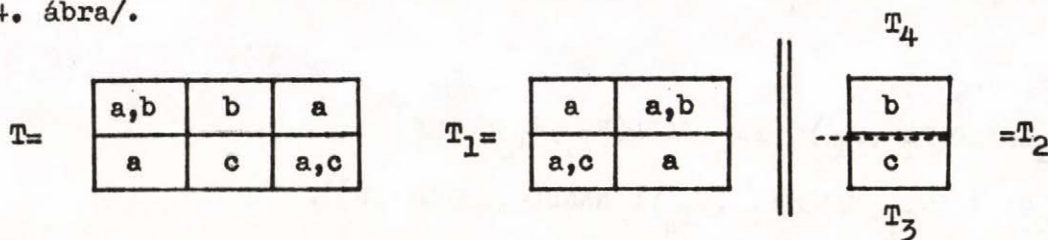
23. ábra

Megjegyzés: Az algoritmus  $\{NOR\}$  bázisbeli szintézist ad, ha a  $\neg T$  oszlop-irányu felbontásait tekintjük.

A  $T$  és  $\neg T$  táblázatok közötti kapcsolatot a III.2.-ben elemeztük. Ennek alapján megvizsgálva az algoritmus lépéseit, világos, hogy a  $\neg T$  táblázat előállítása elkerülhető a következő szabályok alkalmazásával.

1. A  $T$  táblázatot először oszlopirányu résztáblázatokra bontjuk. /A résztáblázatok további felbontása tetszés szerint történhet - ugyanugy mint a III.3.-ban./
2. Ha egy változó egy sorirányu felbontással kapott résztáblázattal kompatibilis, akkor ő maga, egyébként a negáltja szerepel a bázisfüggvény argumentumaként.

Az előző példabeli függvény szintézise ennek megfelelően egyszerűsödik /24. ábra/.



24. ábra

$$F_{T_1} = \bar{a}, \quad F_{T_3} = c, \quad F_{T_4} = b.$$



## Szintézis bizonyos korlátozások figyelembevételével és a határok felhasználásával

E megfontolások figyelembevételével a III.3.-ban kidolgozott algoritmus minimális változtatásával olyan algoritmust dolgoztunk ki  $\Phi$ -Boole-függvények  $\{\text{NAND}\}$  bázisbeli szintézisére, amely figyelembe veszi a bázisfüggvény változóinak maximális számára, valamint a zárójelmélységre tett korlátozásokat.

Az algoritmussal kapcsolatos jelölések azonosak a III.3.-ban leírt algoritmus jelöléseivel, mindössze  $m(1)$ ,  $m(2)$  helyett  $m$  jelöli a bázisfüggvény változóinak maximális számát.

### Algoritmus

1.  $i \leftarrow 1$ ;  $a \leftarrow 0$ ;  $z \leftarrow 0$ ;  $F \leftarrow 0$ ;  $mq \leftarrow 0$ ;  
 $H_1(k, l, lz) \leftarrow f_{T_1}, \hat{f}_{T_1}^* : k \leftarrow 1; l \leftarrow 1; lx \leftarrow 1; KEZD \leftarrow 0$ .
2. Ha  $mz=2$ , akkor az algoritmus a 27. lépéssel folytatódik.
3. Kiválasztjuk a  $H_1$ -ben maximális súlyu  $x_j$ -t.
4. Ha  $H_1$  kompatibilis a max. súlyu  $x_j$ -vel, akkor a 20. lépés következik.
5. Ha  $KEZD=0$ , akkor  $KEZD \leftarrow 1$  és a 18. lépés következik.
6.  $z \leftarrow z+1$ ,  $z \leftarrow z+1$ ;  $F \leftarrow ($ .
7. Ha  $la=0 \equiv l=1$ , akkor a 16. lépés következik.
8. Ha  $p \neq 2$  vagy  $a \neq 0$  nem teljesül, akkor a 17. lépés következik.
9. Ha  $lx=1$ , akkor  $k \leftarrow 2$  és a 11. lépés következik.
10.  $k \leftarrow k+1$
11.  $l \leftarrow 2$ ,  $H_1$  sorirányu felbontása  $H_1, H_{1+1}$ -re;  $i \leftarrow i+1$ ;  $l \leftarrow 2$ .
12.  $k \leftarrow 1$ ;  $lx \leftarrow l$ ;  $lz \leftarrow 0$ .
13. Ha  $x_1$  kompatibilis  $H_1$ -vel, akkor 20. lépés következik.
14. Ha  $z=mz$ , akkor a 27. lépés következik.
15.  $l \leftarrow l-1$  és a 3. lépés következik.
16. Ha  $p \neq 1$  vagy  $la=0$  nem teljesül, akkor a 9. lépés következik.
17. Ha  $lx=1$ , akkor  $k \leftarrow k+1$  és a 19. lépés következik.

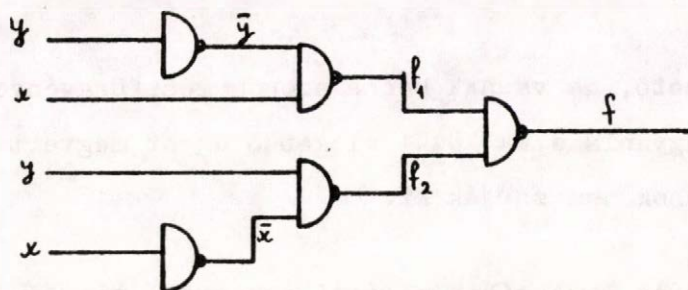


- ### III.5. Általános Boole-függvények szintézise a határok összehasonlítási módszerével $\{\wedge, \vee, \neg\}$ és $\{\text{NAND}\}$ bázisban

A III.2.-ben adott táblázat-módszer vizsgálatával megállapítható, hogy egy Boole-függvény szintézisét felfoghatjuk a résztáblázatokhoz tartozó rész-

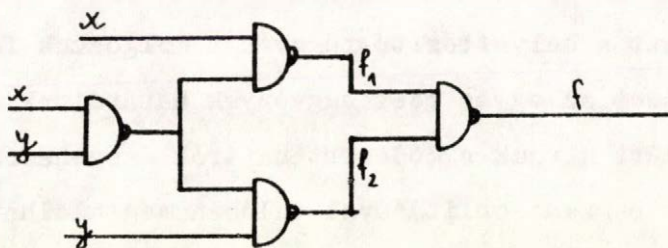


függvényekből, mint komponens függvényekből álló általános Boole-függvény szintézisének is. Az  $f = x \oplus y$  függvény NAND bázisbeli szintézise a III.4. első részében leírt módszerrel pl. az  $\{f_1 = \bar{x} \vee y, f_2 = x \vee \bar{y}\}$  általános Boole-függvény szintézisét jelenti. A szintézis eredménye alapján kapjuk az alábbi hálózatot.



25. ábra

A hálózatot megvizsgálva látjuk, hogy a funkcionális elemek száma eggyel csökkenthető, mivel  $f_1$ -et ill. az  $f_2$ -t realizáló funkcionális elemek  $\bar{x}$   $\bar{y}$  bemenetei helyettesíthetők  $\text{NAND}(x,y)$ -al. Így az  $f$ -et realizáló előző hálózat az alábbira egyszerűsödik /26. ábra/.



26. ábra

A függvényt realizáló konkrét hálózat ismerete nélkül az ilyen egyszerűsítési lehetőségeket nehéz észrevenni. Mint látni fogjuk, az alábbiakban kidolgozott módszerrel az ilyen jellegű egyszerűsítések felismerhetők.



### A határok összehasonlításának módszere

A III.3. és a III.4. pontbeli algoritmusokban a T táblázat szétvágását a táblázathoz tartozó függvény alsó és felső határainak felbontásával helyettesítettük. A szintézis eredményeként kapott formulát realizáló hálózatban az egyes funkcionális elemek egy-egy résztáblázattal kompatibilis teljesen meghatározott Boole-függvényt realizálnak.

A hálózat egyszerűsíthető, ha vannak benne azonos részfüggvényeket realizáló funkcionális elemek, ugyanis ezek közül elegendő egyet megtartani, ha ezt technológiai korlátozások nem zárják ki.

Mivel az algoritmus egyes lépéseiben a részfüggvények, mint  $\Phi$ -Boole-függvények, alsó és felső határakkal adottak, ezért a közös részfüggvények megtalálására nagyobb lehetőség nyílik, ha nem a szintézis eljárás végén kapott teljesen meghatározott függvények között keressük azokat, hanem a szintézis eljárás közben. Ezért az eljárás során a részfüggvényekként kapott  $\Phi$ -Boole-függvényeket megvizsgáljuk abból a szempontból, hogy van-e olyan teljesen meghatározott vagy  $\Phi$ -Boole-függvény, amelyik kompatibilis két- vagy több részfüggvénnyel; ha igen, akkor azokat a részfüggvényeket helyettesítjük az így megtalált függvénnyel. A szintézis következő lépésében az eredeti részfüggvények helyett ezt a helyettesítő függvényt dolgozzuk fel. A közös részfüggvények megkeresését az egyes részfüggvények határainak összehasonlításával oldjuk meg /ezért hívjuk a módszert határok összehasonlítása módszerének/. Hogy a határok összehasonlításával valóban megtalálhatók a közös részfüggvények, az előbb megvizsgált  $f = x \oplus y$  függvény példáján szemléltetjük.

$$f = x \oplus y = x\bar{y} \vee \bar{x}y$$

$$\bar{f} = f_1 f_2$$

$$f_1 = \bar{x}\bar{y} \vee xy \vee y(\bar{x} \vee y)$$

$$f_2 = \bar{x}\bar{y} \vee xy \vee y(x \vee \bar{y})$$

$$\bar{f}_1 = f_3 f_4$$

$$f_3 = \bar{x}\bar{y} \vee y(\bar{x} \vee \bar{y})$$



$$f_4 = x\bar{y} \vee y(x \vee y)$$

$$\bar{f}_2 = f_5 f_6$$

$$f_5 = \bar{x}y \vee y(\bar{x} \vee \bar{y})$$

$$f_6 = \bar{x}y \vee y(x \vee y)$$

$f_4$  ill.  $f_6$  kompatibilis  $x$  ill.  $y$ -nal, tehát további szintézisre nincs szükség.  $f_3$  ill.  $f_5$  kompatibilis  $\bar{y}$  ill.  $\bar{x}$ -tal, tehát a szintézisük nem fejeződött be.  $f_3$  ill.  $f_5$  alsó határa  $x\bar{y}$  ill.  $\bar{x}y$ , felső határuk azonos  $\bar{x} \vee \bar{y}$ . A felső határ realizálható egyetlen NAND funkcionális elemmel, ezért  $f_3 = f_5 = \text{NAND}(x, y)$ .

Igy  $f = \text{NAND}(f_1, f_2) = \text{NAND}(\text{NAND}(f_3, f_4), \text{NAND}(f_5, f_6)) = \text{NAND}(\text{NAND}(\text{NAND}(x, y), x), \text{NAND}(\text{NAND}(x, y), y))$ , amely a 26. ábrán lévő hálózattal realizálható.

Az alábbiakban a III.3. és a III.4.-ben leírt algoritmusok továbbfejlesztett változatait ismertetjük, amelyek a részfüggvények határainak összehasonlításával keresik a közös részfüggvényeket.

Az algoritmusokban általában ugyanazokat a jelöléseket használjuk, mint a megfelelő III.3. vagy III.4. pontban leírt algoritmusokban; az eltérések a következők.  $H_1(l, t, k)$ -ban  $i$  azt mutatja, hogy  $H$ -ból  $H_1$ -t  $i$  db váltakozó irányu felbontással kaptuk,  $t$  azt, hogy a  $H_1$ -nek megfelelő  $\Phi$ -Boole-függvény szintézise megtörtént-e már,  $k$  pedig azt, hogy  $H_1$  hányszor szerepel közös függvényként.  $w(1)$  jelöli a  $\vee$ ,  $w(2)$  az  $\wedge$ ,  $w$  a NAND funkcionális elem kimenetének terhelhetőségét.

Algoritmus  $\{\wedge, \vee, \neg\}$  bázisbeli szintézisre a határok összehasonlításának módszerével

1.  $i \leftarrow 1; l \leftarrow 2$ .
2. Minden  $H_1(l, 0, k)$  felbontása  $2 \leq K \leq m$   $3-l$  részre  $3-l$  által meghatározott irányban;  $i \leftarrow i+1; l \leftarrow 3-l; t \leftarrow 0; v \leftarrow 0$ .
3.  $v \leftarrow v+1$ ; Ha  $H_1$  kompatibilis egyetlen változóval, akkor  $H_1$  szintézise



befejeződött és a 10. lépés következik.

4. Ha van  $H_1(\ell, 0, s)$ -el kompatibilis  $H_j(\ell, t, k)$ ,  $1 \leq j \leq i$ , akkor a 8. lépés következik.
5. Ha nincs  $H_1(\ell, 0, s)$ -hez  $H_j(\ell, 0, k)$ ,  $1 \leq j \leq i$ , hogy  $\hat{f}_{T_1} \leq \hat{f}_{T_j}$  és  $\hat{f}_{T_1} \geq \hat{f}_{T_j}$ , vagy  $\hat{f}_{T_1} \geq \hat{f}_{T_j}$  és  $\hat{f}_{T_1} \leq \hat{f}_{T_j}$  akkor a 13. lépés következik.
6. Ha  $k+s \geq w(\ell)$  akkor a 10. lépés következik.
7. A kompatibilis  $H_n$  megtartása;  $k \leftarrow k+s$ ; a 10. lépés következik.
8. Ha  $k+s \geq w(\ell)$ , akkor  $H_1 \leftarrow H_j(\ell, t, k)$  és a 10. lépés következik.
9.  $H_1$  azonos  $H_j(\ell, t, k+s)$ -el.
10. Ha  $v < K$ , akkor a 3. lépés következik.
11. Ha van még felbontandó  $H_1(\ell, 0, k)$ , akkor a 2. lépés következik.
12. Az eljárás befejeződik.
13. Ha az  $i$ -hez van olyan  $j$ ,  $1 \leq j \leq i$ , hogy  $H_1(\ell, 0, s)$  és  $H_j(\ell, 0, k)$ -ban  $\hat{f}_{T_1} = \hat{f}_{T_j}$  vagy  $\hat{f}_{T_1} = \hat{f}_{T_j}$  és  $k+s < w(\ell)$ , akkor  $H_1 = H_j = \hat{f}_{T_1}, \hat{f}_{T_1}^*$  vagy  $\hat{f}_{T_1}, \hat{f}_{T_1}^*$
14. 10. lépés következik.

A  $\{\text{NAND}\}$  bázisbeli szintézis sajátosságaiból következik, hogy a közös rész-függvények keresésénél csak az azonos irányú felbontásokhoz tartozó  $H_1$ -ket kell vizsgálni. Egyébként a  $\{\text{NAND}\}$  bázisbeli algoritmus lényegében nem tér el az előzőtől.

#### Algoritmus $\{\text{NAND}\}$ bázisbeli szintézisre a határok összehasonlításának módszerével

1.  $i \leftarrow 1; \ell \leftarrow 2$ .
2. Minden  $H_1(\ell, 0, k)$  felbontása  $2 \leq K \leq m(3-\ell)$  részre  $3-\ell$  által meghatározott irányban;  $i \leftarrow i+1; \ell \leftarrow 3-\ell; t \leftarrow 0; v \leftarrow 0$ .
3.  $v \leftarrow v+1$ ; ha  $H_1$  kompatibilis egyetlen változóval, akkor  $H_1$  szintézise befejeződött és a 10. lépés következik.
4. Ha van  $H_1(\ell, 0, s)$ -el kompatibilis  $H_j(\ell, t, k)$ ,  $1 \leq j \leq i$ , ahol  $\ell$  azonos, akkor a 8. lépés következik.



5. Ha nincs  $H_i(l, 0, s)$ -hez  $H_j(l, 0, k)$ ,  $1 \leq j \leq i$ , hogy  $\hat{f}_{T_i} \leq \hat{f}_{T_j}$  és  $\hat{f}_{T_i} \geq \hat{f}_{T_j}$ , vagy  $\hat{f}_{T_i} \geq \hat{f}_{T_j}$  és  $\hat{f}_{T_i} \leq \hat{f}_{T_j}$ , valamint  $l$  azonos, akkor a 13. lépés következik.
6. Ha  $k+s \geq w$ , akkor a 10. lépés következik.
7. A kompatibilis  $H_n$  megtartása;  $k \leftarrow k+s$ ; a 10. lépés következik.
8. Ha  $k+s \geq w$ , akkor  $H_i \leftarrow H_j(l, t, k)$  és a 10. lépés következik.
9.  $H_i$  azonos  $H_j(l, t, k+s)$ -el.
10. Ha  $v < K$ , akkor a 3. lépés következik.
11. Ha van még felbontandó  $H_i(l, 0, k)$ , akkor a 2. lépés következik.
12. Az eljárás befejeződik.
13. Ha van olyan  $H_i(l, 0, s)$  és  $H_j(l, 0, k)$ ,  $1 \leq j \leq i$ , azonos  $l$ -l, hogy  $\hat{f}_{T_i} = \hat{f}_{T_j}$ , vagy  $\hat{f}_{T_i} = \hat{f}_{T_j}$  és  $k+s < w$ , akkor  $H_i = H_j = \hat{f}_{T_i}, \hat{f}_{T_i}^*$  vagy  $\hat{f}_{T_i}, \hat{f}_{T_i}^*$
14. A 10. lépés következik.

E két utóbbi algoritmus alkalmas általános Boole-függvények szintézisére is.

Ezen algoritmusok azonban általános Boole-függvények esetén mindig keresik a közös részfüggvényeket függetlenül attól, hogy egyes komponens függvények ezt a keresést feleslegessé teszik. Ezért ezen algoritmusok általános Boole-függvényekre való alkalmazása nem gazdaságos; az algoritmusok ilyen irányú tökéletesítése és tetszőleges bázisbeli szintézist megvalósító algoritmusok kidolgozása a következő évek feladatát képezik.







1. A tanulmány célja és feladata
2. A tanulmány módszere
3. A tanulmány eredménye
4. A tanulmány következtetése
5. A tanulmány értékelése
6. A tanulmány összefoglalása
7. A tanulmány zárkövetése
8. A tanulmány értékelése

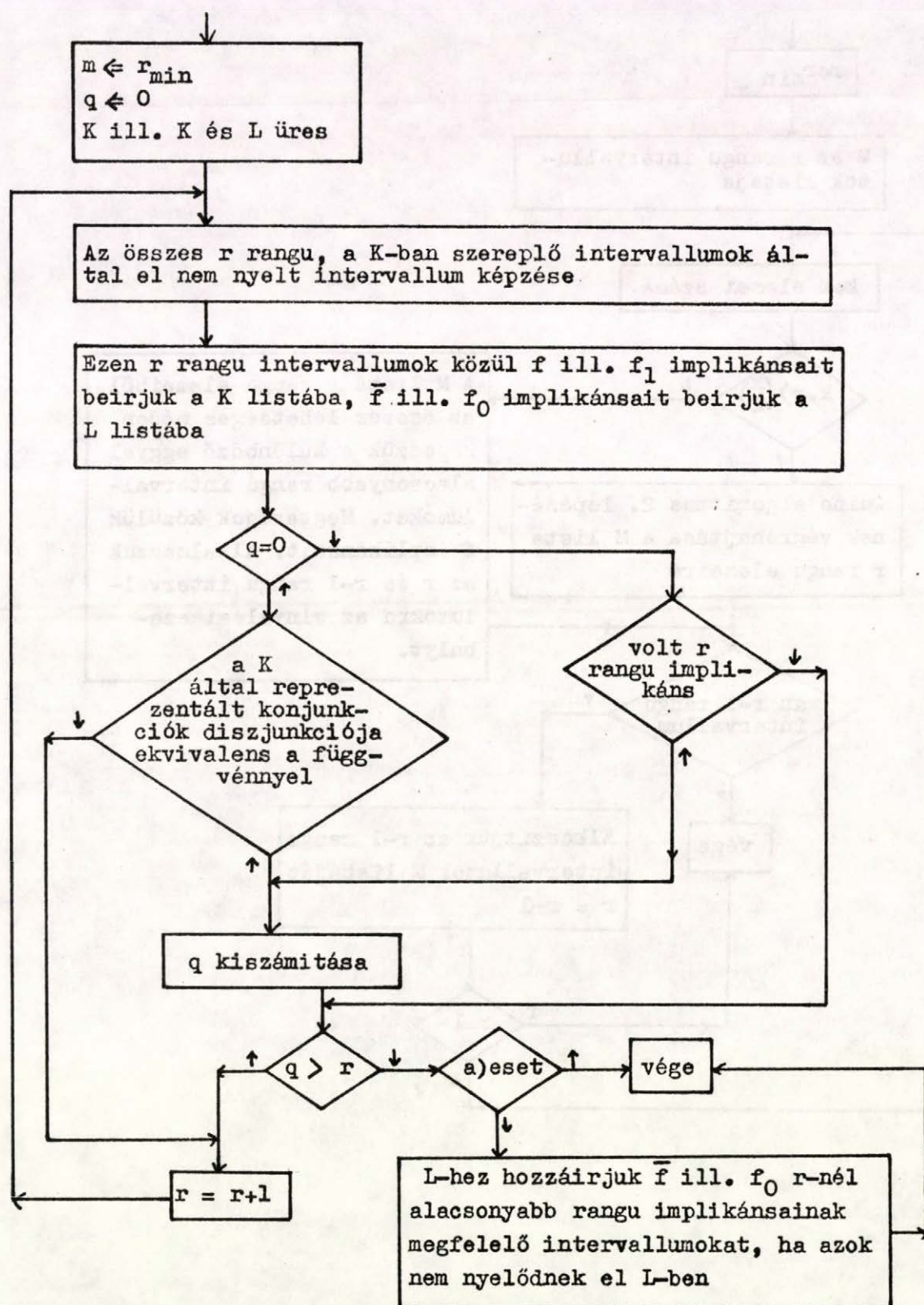
## FÜGGELÉK



### Rövidítések

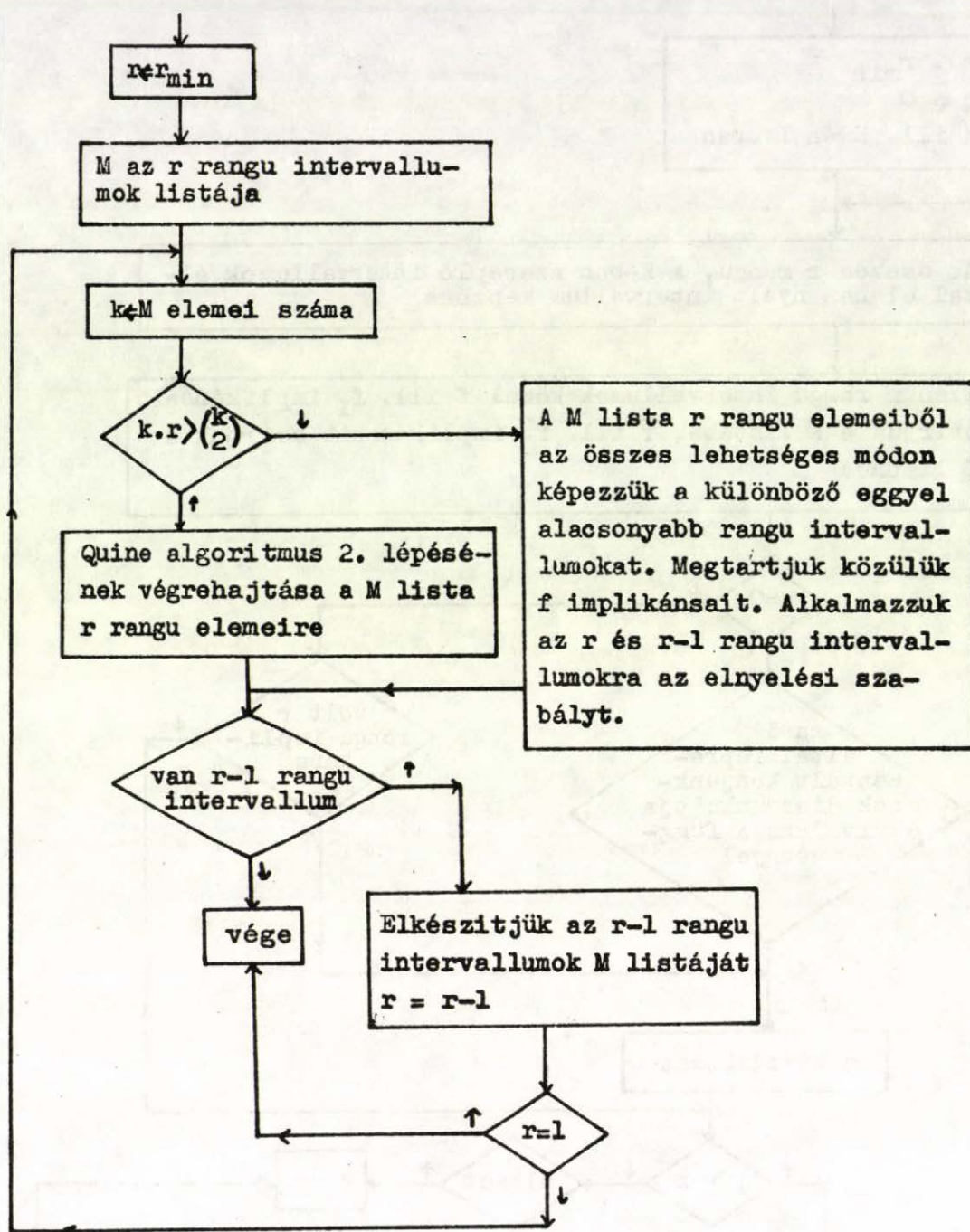
1. KDNF - kitüntetett diszjunktív normálforma /6. definíció/
2. DNF - diszjunktív normálforma /7. definíció/
3. RDNF - redukált diszjunktív normálforma /12. definíció/
4. IDNF - irredundáns diszjunktív normálforma /13. definíció/
5. MDNF - minimális diszjunktív normálforma /14. definíció/
6. KNF - konjunktív normálforma
7. KKNF - kitüntetett konjunktív normálforma
8.  $\Phi$ -Boole-függvény - nem teljesen meghatározott Boole-függvény





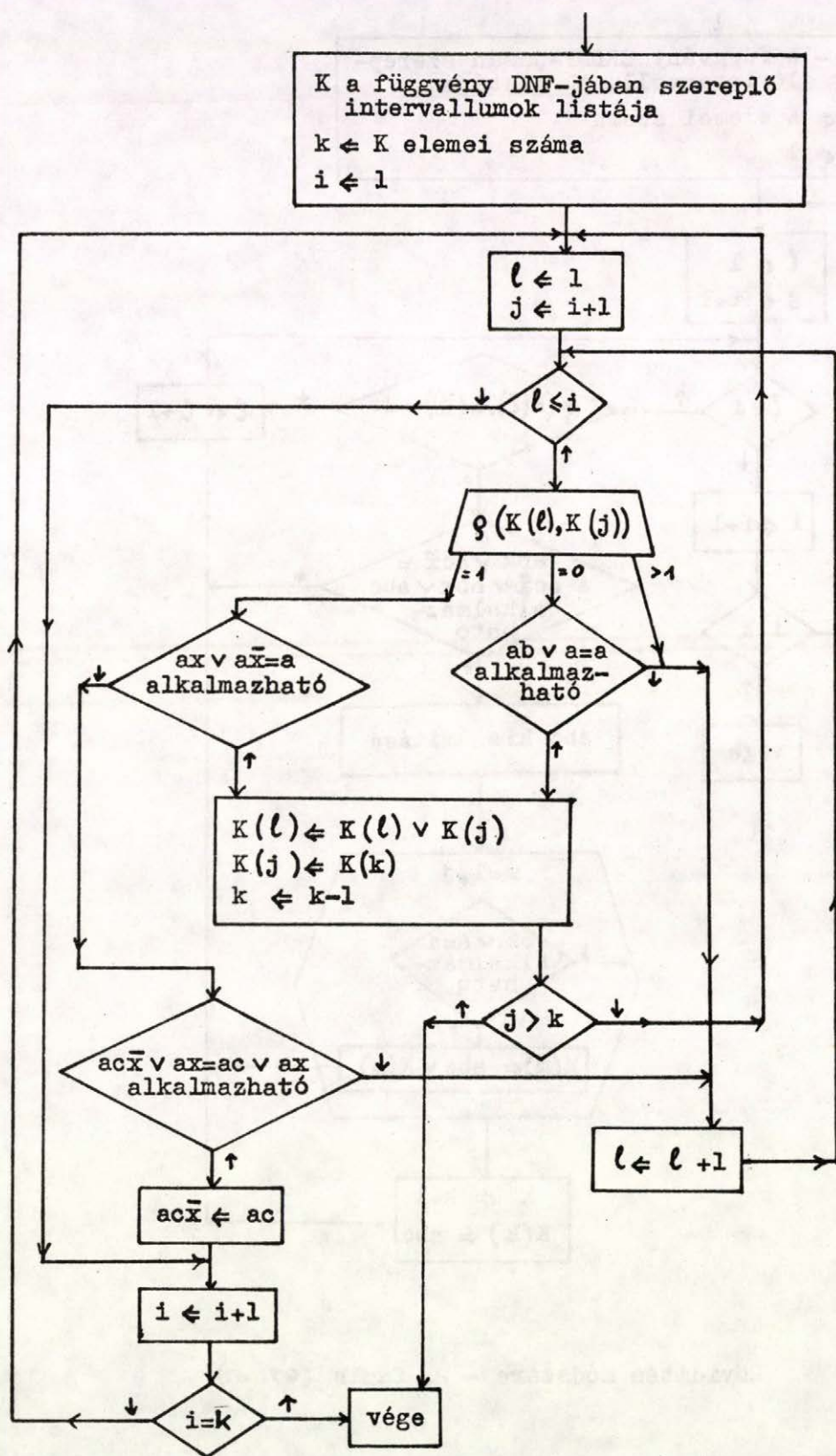
Kipróbálás módszere 1. fázis (64. o.)





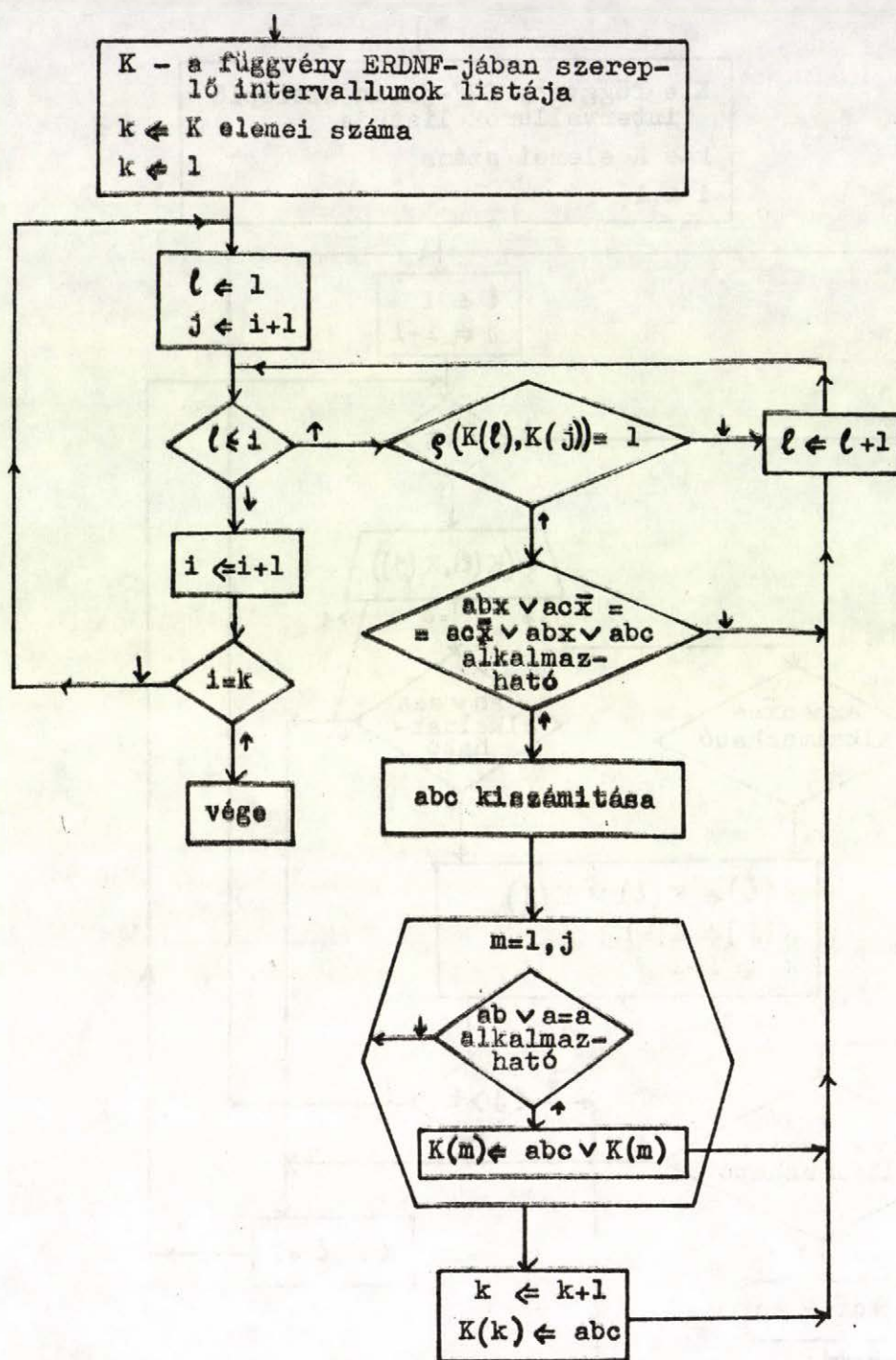
Kipróbálás módszere 2. fázis (66. o.)





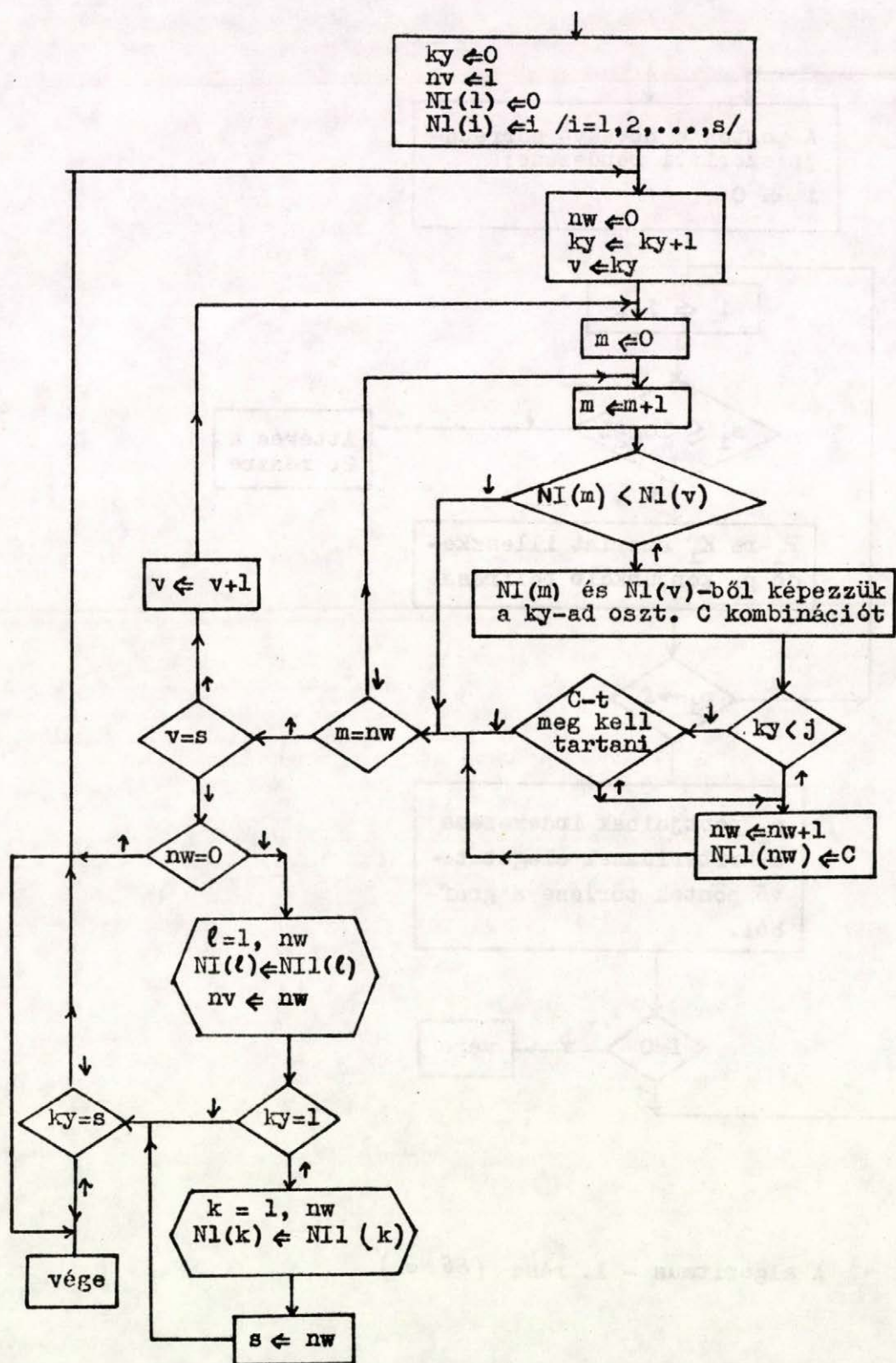
Rövidítés módszere - 1. fázis (68. o.)





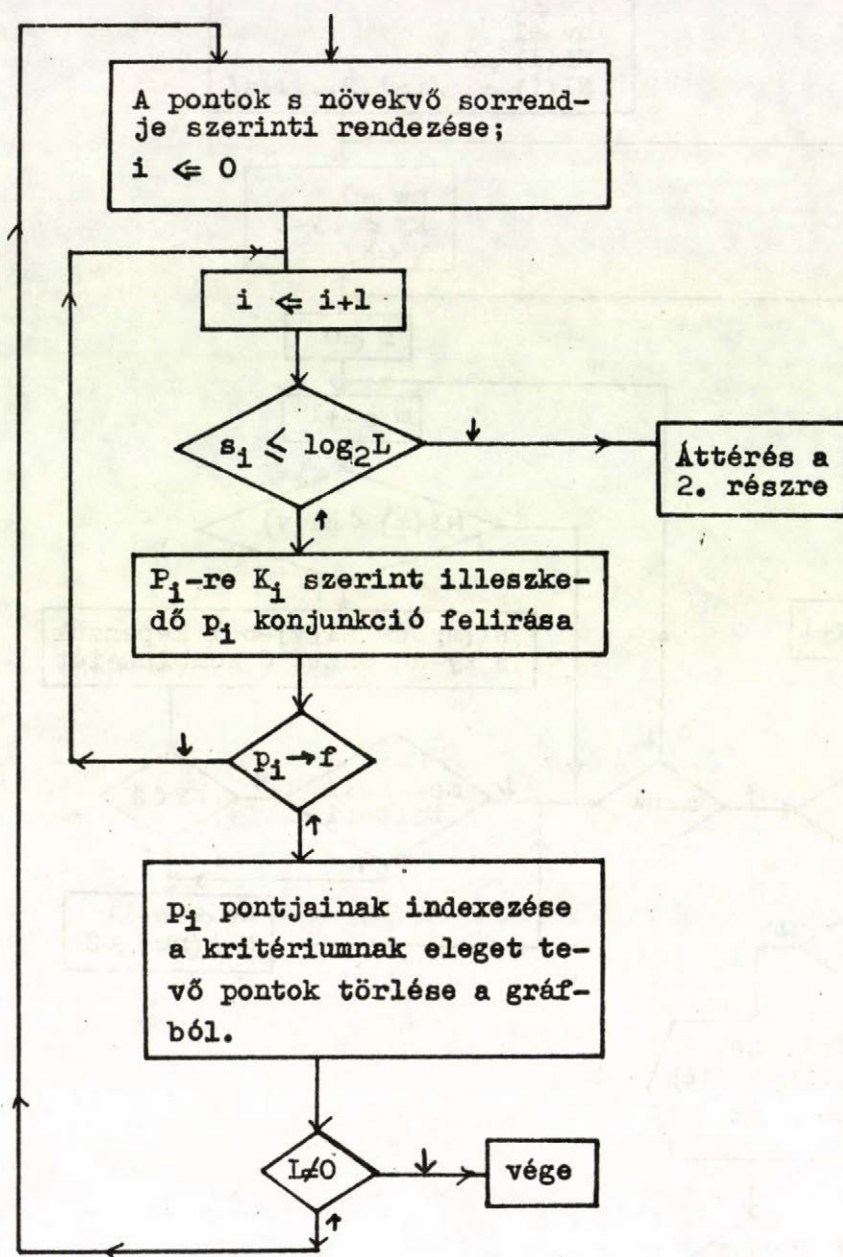
Rövidítés módszere - 2. fázis (69. o.)





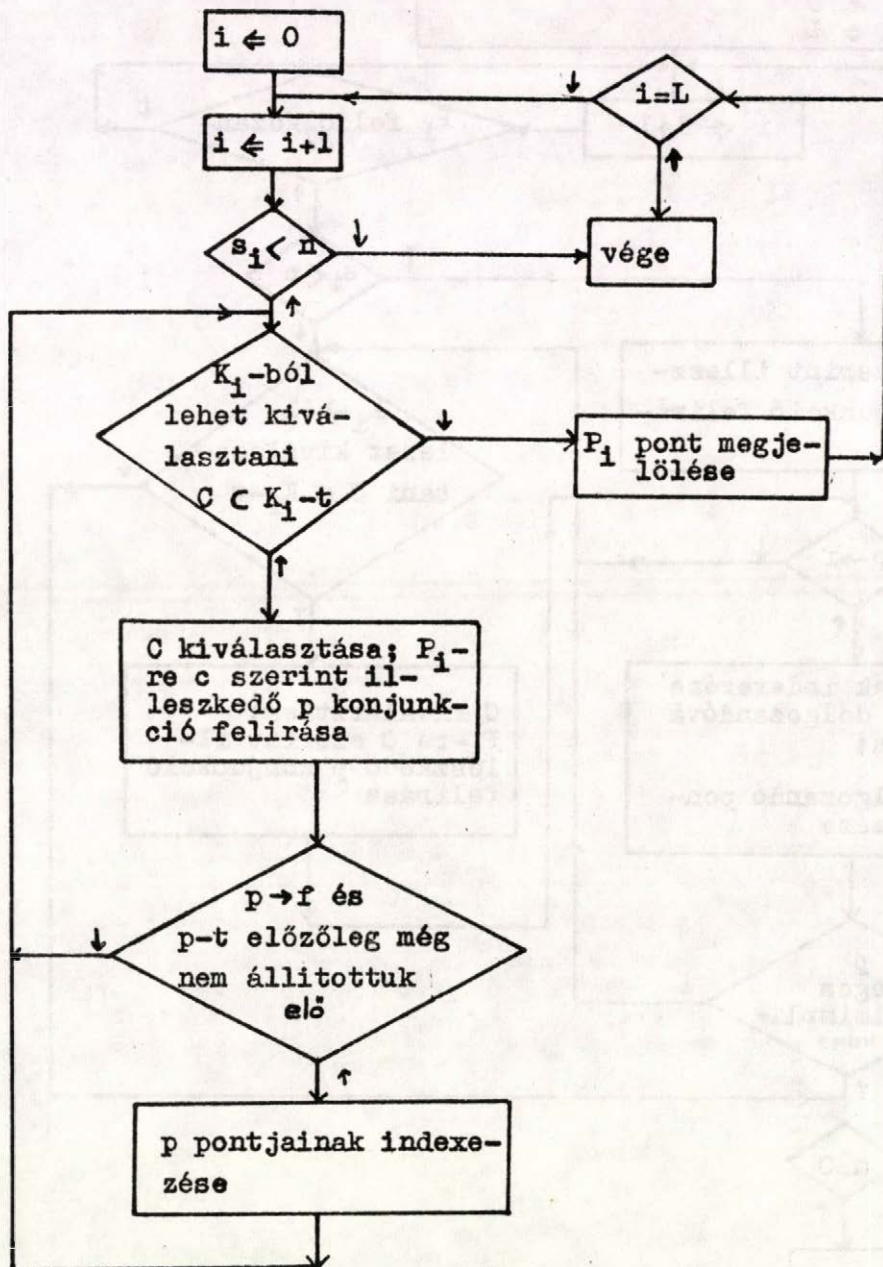
$n$  elem  $v$ -ed osztályu kombinációinak képzése (84. o.)





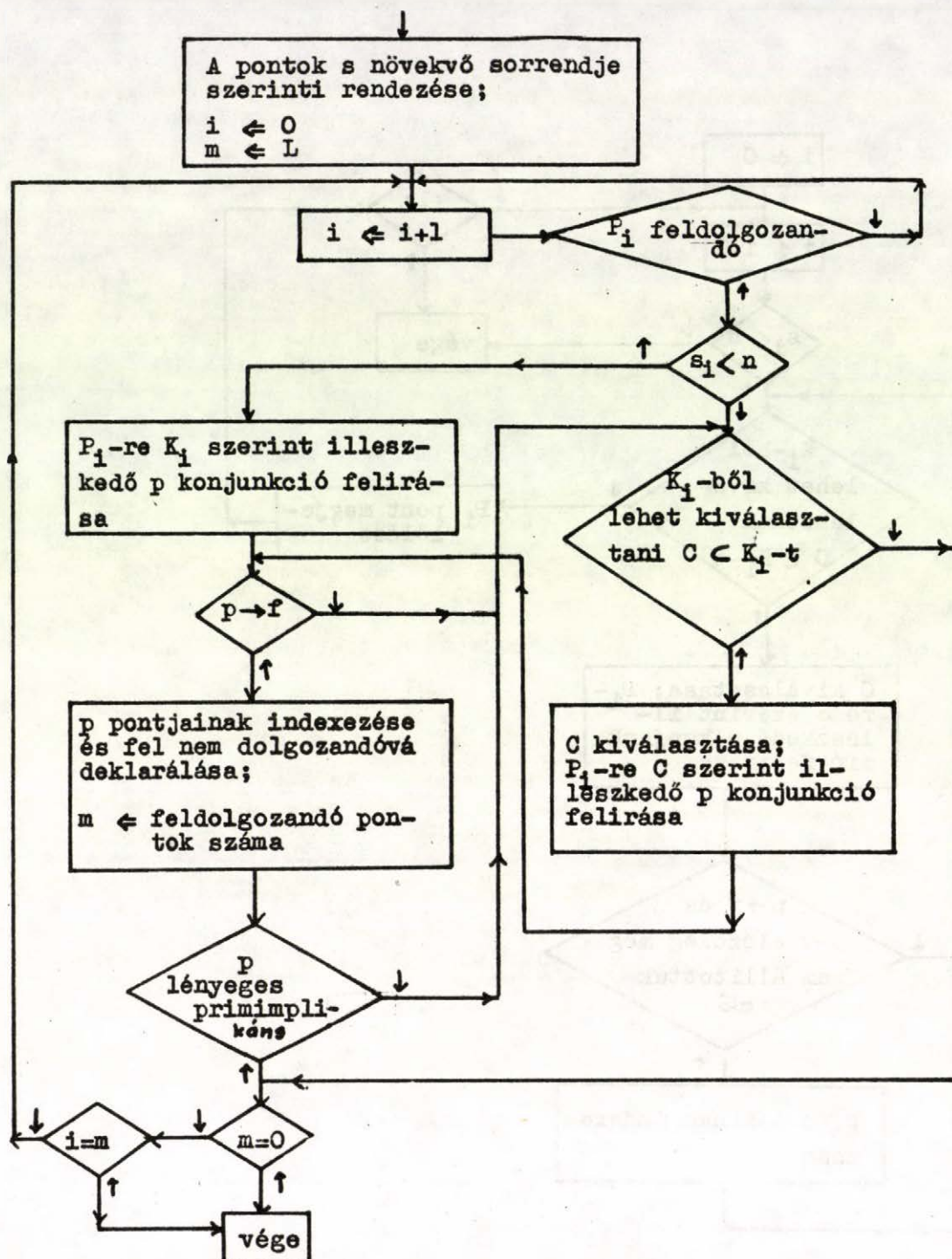
A algoritmus - 1. rész (86. o.)





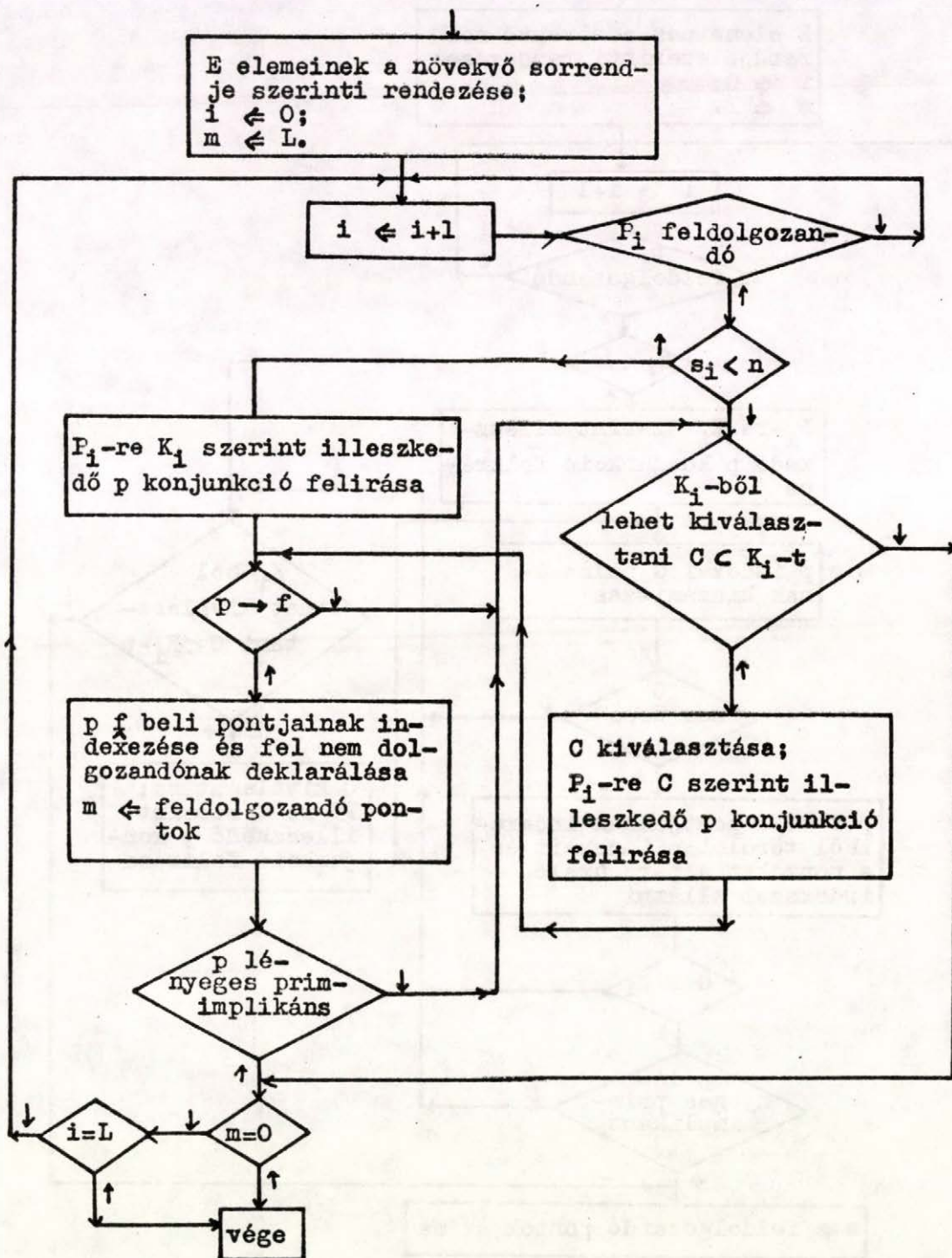
A algoritmus 2. rész (87.o.)





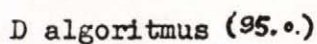
B algoritmus (88. o.)



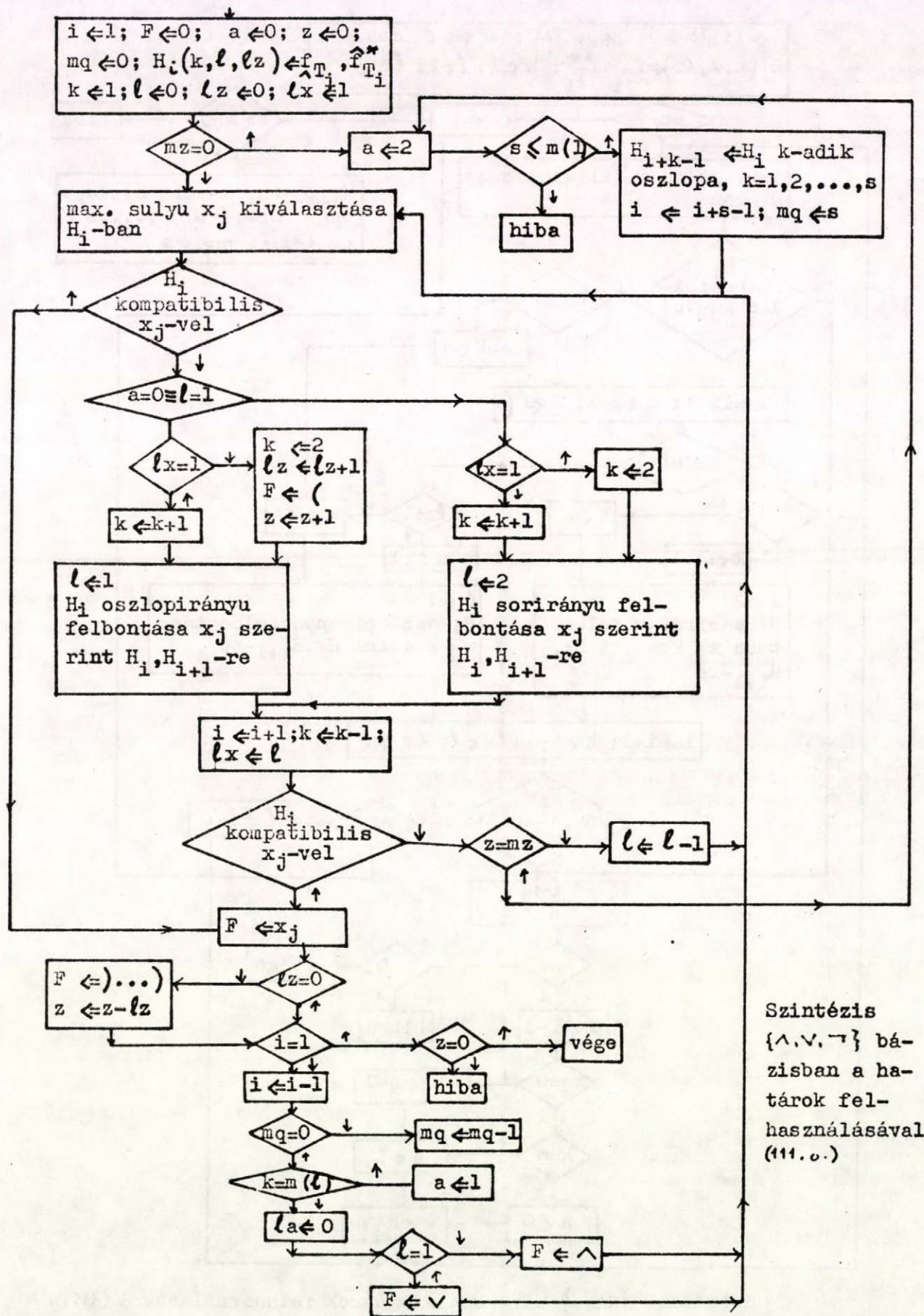


C algoritmus (94. o.)

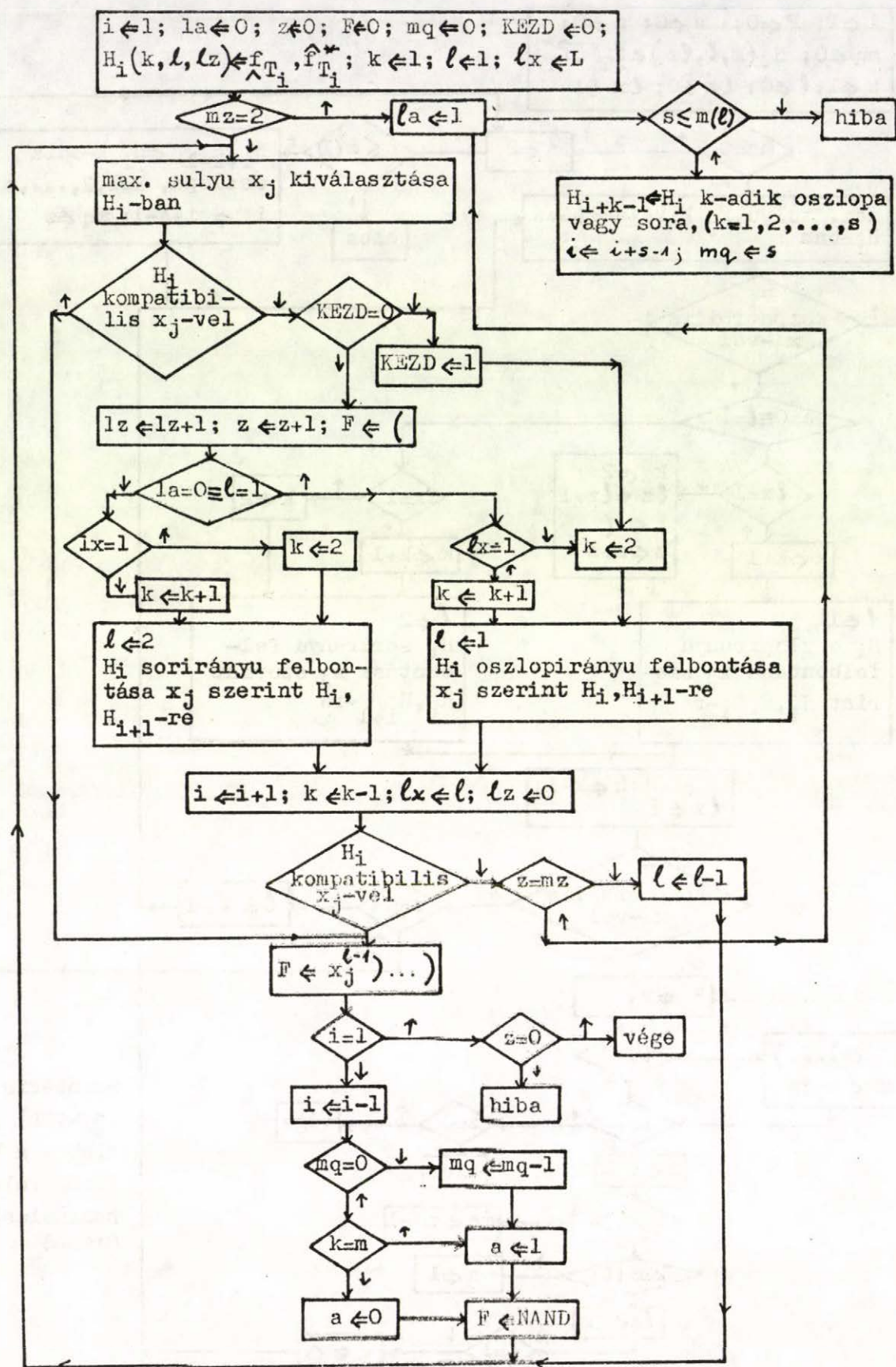






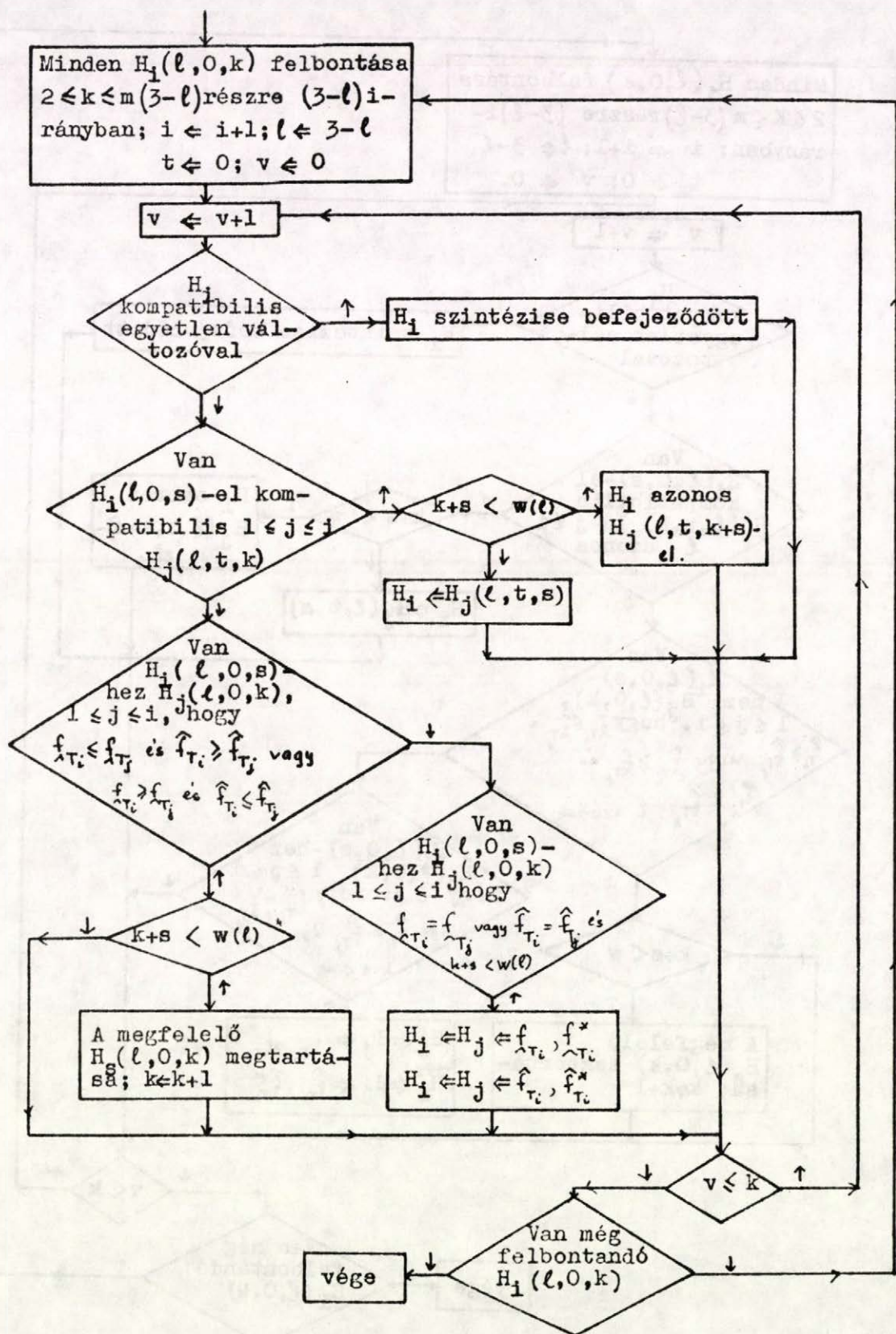






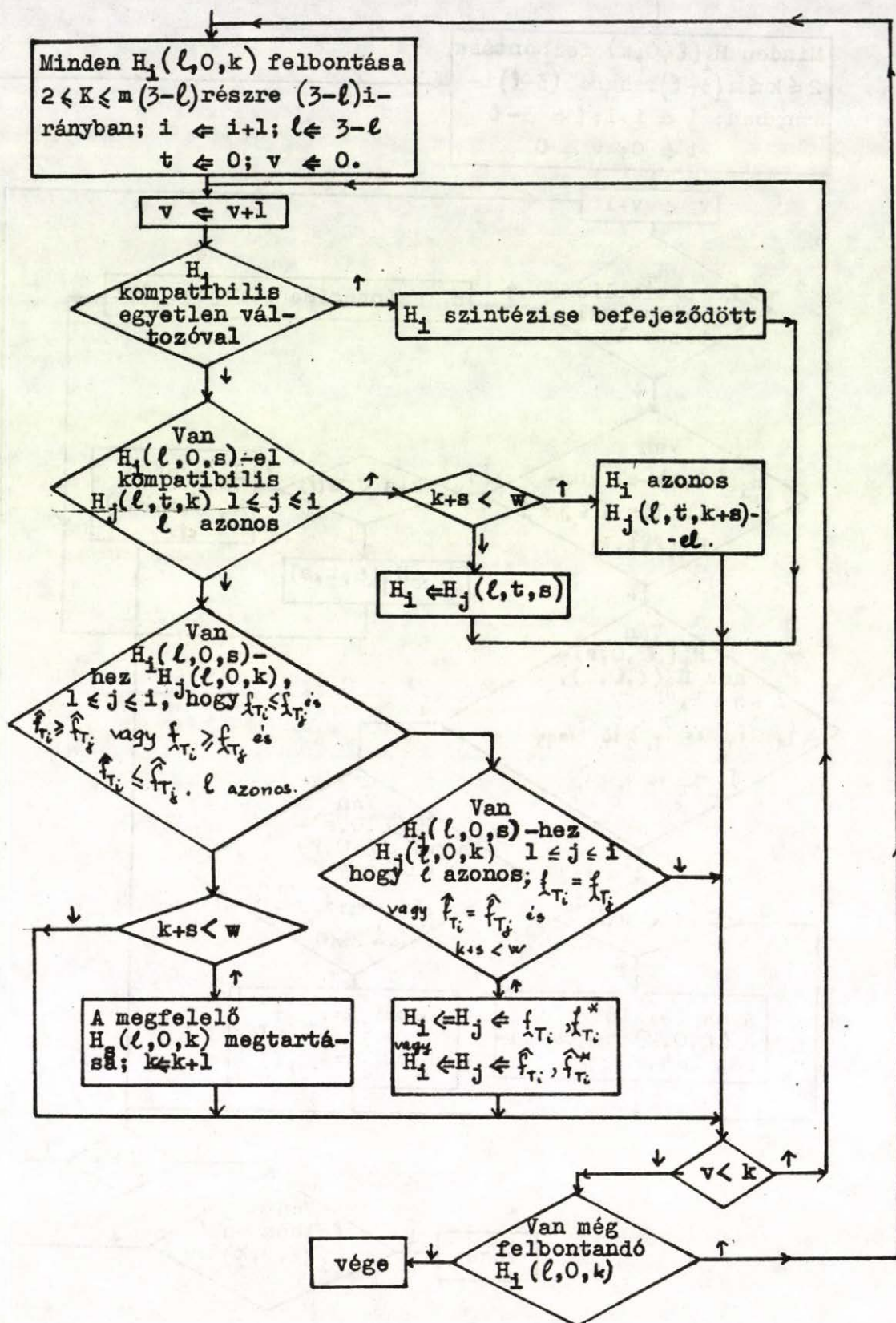
Szintézis {NAND} bázisban a határok felhasználásával (445.o.)





Általános Boole-függvény szintézise  $\{\wedge, \vee, \neg\}$  bázisban a határok összehasonlításával (19. o.)





Általános Boole-függvény szintézise  $\{\text{NAND}\}$  bázisban a határok összehasonlításával. (120. o.)



# IRODALOMJEGYZÉK

- 1 S. Abhyankar: Minimal "Sum of Product of Sums" Expression of Boolean Funktion. IRE Transactions on EC. Vol. EC-7. № 12, Dec. 1958. 268-276.
- 2 Aiken, H.H., and Staff of the Computation Laboratory, Harvard Univ., Synthesis of Electronic Computing and Control Circuits, Harvard University Press, Cambridge, Mass., 1951.
- 3 М.А.Айзерман, Л.А.Гусев, Л.И.Розоноэр, И.М.Смирнова, А.А.Таль: Логика, автоматы, алгоритмы. Физ.Мат.Лит. Москва, 1963.
- 4 S.B. Akers: A Truth Table Method for the Synthesis of Combinational Logic. IRE Transactions on EC. Vol. EC-10. № 12, December 1961. 604-615.
- 5 S.B. Akers: A diagrammatic Approach to Multilevel Logic Synthesis. IEEE Transactions on EC. Vol. EC-14. № 4. Apr. 1965. 171-181.
- 6 P.Sz. Alekszandrov: Bevezetés a halmazok és függvények általános elméletébe. Akadémiai Kiadó, Budapest, 1952.
- 7 D.D. Aufenkamp and F.E. Hohn: Analysis of Sequential Machines I. IRE Transactions on EC. Vol. EC-6. № 12. Dec. 1957. 276-285.
- 8 D.D. Aufenkamp: Analysis of Sequential Machines II. IRE Transactions on EC. Vol. EC-7. № 12. Dec. 1958. 299-306.
- 9 A. Ádám: Truth Functions and the problem of their realization by two terminal graphs. Akadémiai Kiadó, Budapest, 1968.
- 10 Ádám, A.: The quasi-series decomposition of two-terminal graphs, Publicationes Math. /Debrecen/ 10, 1963. 96-107.
- 11 Ádám, A.: Corrections to my paper "The quasi-series decomposition of two-terminal graphs", Publicationes Math. /Debrecen/.
- 12 T.C. Bartee: Computer Design of Multiple-Output Logical Networks. IRE Transactions on EC. Vol. EC-10. № 3. March. 1961. 21-30.
- 13 В. Боднарчук, Л.А.Калужнин, В.Н.Котов, Б.А.Ромов: Теория галуа для алгебр поста. I. Кибернетика №3 май-июнь 1969. 1-10.



- 14 В.Г.Боднарчук, Л.А.Калужнин, В.И.Котов, Б.А.Ромов: Теория галуа для алгебр поста. II. Кибернетика №5 сентябрь-октябрь 1969. I-9.
- 15 A. Bouchet: An Algebraic Method for Minimizing the Number of States in an Incomplete Sequential Machine. IEEE Transactions on Computers, Vol. C-17. № 8. Aug. 1968. 795-798.
- 16 M. Carvallo: Monographie des treillis et algèbre de Boole. Gauthier-Villars, Paris, 1966.
- 17 E.J. McCluskey, Jr. and H. Shorr: Essential Multiple-Output Prime Implicants. Microwave Res. Inst. Symp. ser. 12. 1962. 437-457.
- 18 E.J. McCluskey, Jr.: Minimum-State Sequential Circuits for a Restricted Class of Incompletely Specified Flow Tables. Bell Syst. Techn. J. Vol. 41. Dec. 1962. 1759-1768.
- 19 E.J. McCluskey, Jr.: Minimization of Boolean functions. Bell System Technical Journal Vol. 35. № 6. Nov. 1956. 1417-1444.
- 20 Csendes J., Ivánka G.: Logikai hálózatok számítógépes tervezése. Számítógép alkalmazása a gépiparban konferencia 1969. Miskolc.
- 21 M. Denouette, E. Daclin, J.P. Perrin, N. Bréaud: Décomposition des Fonctions Booléennes. Automatisme Tome XIV. № 5. Mai 1969.
- 22 Fenyő - Frey: Matematika villamosmérnököknek I. Műszaki Könyvkiadó, Budapest, 1964.
- 23 J. Florine: Le synthèse des machines logiques. Dunod, Paris, 1964.
- 24 L. Frécon: Écriture atomique des fonctions booléennes. Automatisme. Tome XVI. № 6-7. Juin-Juillet 1971. 341-343.
- 25 Frey Tamás: Automaták, algoritmusok, optimalizálásuk és approximációjuk. Doktori disszertáció, 1970.
- 26 Г.Р.Фрицнович: Минимизация числа состояний частично определенных асинхронных автоматов. Автоматика и вычислительная техника, 1970. 4. I-7.
- 27 М.А.Гаврилов: Минимизация булевых функций, характеризующих релейных цепи. Автоматика и телемеханика, 1967. №9.



- 28 Gécseg Ferenc és Peák István: Az automaták algebrai elmélete.  
Matematikai Lapok 17, 1966. 77-134.
- 29 S. Ginsburg: An Introduction to Mathematical Machine Theory.  
Reading, Mass.: Addison-Wesley, 1962.
- 30 A. Grasselli and F. Luccio: A Method for Minimizing the Number of  
Internal States in Incompletely Specified Sequential Net-  
works. IEEE Transactions on EC. Vol. EC-14. June 1965.  
350-359.
- 31 М.С.Гриншпон: Синтез комбинационных логических устройств на  
элементах или-не (и-не). Приборы и устройства вычисли-  
тельной техники, 1971. №4. 49-53.
- 32 B. Harris: An Algorithm for Determining Minimal Representations of  
of a Logic Function. IEE Transactions on EC. Vol. EC-6. № 6.  
June 1957. 103-108.
- 33 И.Г.Илзиня, Г.Ф.Фрицнович: Кодирование внутренних состояний  
асинхронного конечного автомата п-кодом. Автоматика и  
вычислительная техника, 1970. №6. 5-II.
- 34 M. Karnaugh: The Mar Method for Synthesis of Combinational Logic  
Circuits. Trans. AIEE Communications and Electronics Nov.  
1953. 593-599.
- 35 В.Д.Казаков: О скобочных минимальных выражениях булевых  
функций.  
Deuxième Congrès Mathématique Hongrois, Budapest, 1961. Vol.2.
- 36 J. Kenneth Butler Jr., J.N. Warfield: A digital computer program  
for reducing logical statements to a minimal form. Proc.Natl.  
Electronics Conf. 1959. 456-466.
- 37 J. Kuntzmann: Algèbre de Boole. Dunod, Paris, 1968.
38. R.S. Ledley: Digital Computer and Control Engineering McGraw-Hill,  
New York, London, 1960.
39. C.Y. Lee: Switching functions on an N-dimensional cube. Communica-  
tions and Electronics № 14. Sept. 1954.



- 40 H.M. Lipp: Der Entwurf von Schaltnetzen mit NAND/NOR-Gliedern  
Elektron. Rechenanl. 13/1971/ H.2. 65-74.
- 41 A.R. Meo: On the Minimal third order Expression of a Boolean Function. AIEE Proc. 3. Symp. on SCT and L.D. 1962. pp. 5-24.
- 42 E.F. Moore: Gedanken-experiments on Sequential Machines. Automata Studies. Princeton, New Jersey, 1956. 129-153.
- 43 F. Móríc: On a minimization algorithm for Boolean functions. Acta Cybernetica, Tom 1. F.2. 1971. Sept. 97-105.
- 44 E. Morreale: Rekursive Operators for Prime Implicant and Ir-redundant Normal Form Determination. IEEE Transactions on Computers Vol. C-19. № 6. June 1970. 504-509.
- 45 H. Mott, C.C. Carroll: Numerical Procedures for Boolean Function Minimization. IEEE Transaction on EC. Vol. EC-13. № 4. Aug. 1964. 470-471.
- 46 D.E. Muller: Application of Boolean algebra to switching circuit design and to error detection. IRE Trans. Vol. EC-3. Sept. 1954. 6-12.
- 47 R. McNaughton, B. Mitchell: The Minimality of Rectifier Nets with Multiple Outputs Incompletely Specified. Journal of Frankl. Inst. Vol. № 6. 1957. 457-479.
- 48 Н.Н. Некула: Автоматическая минимизация двузначных булевых функций. Автоматика и телемеханика, 1967. №5. 80-85.
- 49 R. Novansky: Syntéza sekvenčních obvodu. Kybernetika. Číslo 3, Ročník 3/1967. 234-252.
- 50 Pásztorné, Varga Katalin: Nem teljesen meghatározott Boole-függvények szintézise  $\{\wedge, \vee, \neg\}$ ,  $\{\text{NAND}\}$  ill.  $\{\text{NOR}\}$  bázisban. MTA Számítástechnikai Központja Közlemények, 1972. 3. szám.
- 51 K. Pásztor - Varga: On some minimizing Algorithms of Boolean Functions. Acta Technica Academiae Scientiarum Hungaricae, Tomus 72 /1-2/, pp. 365-378.
- 52 M.C. Paull and S.H. Unger: Minimizing the Number of States in Incompletely Specified Sequential Switching Functions.



IRE Transactions on EC. Vol. EC-8. № 8. Sept. 1959.

356-367.

- 53 E. Fichat: Decomposition des fonctions booléennes. Thèses. Univ. de Grenoble, 1966.
- 54 Д.А.Поспелов: Логические методы анализа и синтеза схем. Изд. "Энергия", Москва, 1964.
- 55 E.L. Post: Two valued iterative systems of mathematical logic. Annals of math. Studies /5/ Princeton Univ. Press. 1941.
- 56 W.V. Quine: The problem of Simplifying Truth Functions. Amer. Math. Monthly, Vol. 59. October, 1952. 521-531.
- 57 W.V. Quine: A Way to Simplify Truth Functions. Amer. Math. Monthly, Vol. 62. Nov. 1955. № 9. 627-631.
- 58 C. Reischer and Dan A. Simovici: Associative Algebraic Structures in the Set of Boolean Functions and Some Applications in Automata Theory. IEEE Transactions on Computers. Vol. C-20. № 3. March 1971. 298-303.
- 59 J. Richalet: Calcul Opérationnel Booléen. L'Onde Électrique. № 439. Octobre 1963. 1003-1023.
- 60 Szász Gábor: Bevezetés a hálóelméletbe. Akadémiai Kiadó, Budapest 1959.
- 61 P. Tison: Generalisation of Consensus Theory and Application to the Minimization of Boolean Functions. IEEE Transactions on EC Vol. EC-16. № 4. Apr. 1964. 446-456.
- 62 P. Tison: Théorie des consensus. Thèses. Univ. de Grenoble, 1965.
- 63 P. Tison: Recherche des termes premiers d'une fonction Booléenne Applications. Automatisme 1964. № 1. 15-19.
- 64 P. Tison: Fondement d'une Algèbre des Systèmes Logiques. Automatisme. Tome XII. № 7-8. Juillet-Aout 1967. 298-303.
- 65 P. Tison: Une Logique des Machines Séquentielles. Automatisme. Tome XII. № 9. Septembre 1967. 345-351.
- 66 R.H. Urbano, R.K. Mueller: A Topological Method for the Determination of Minimal Forms of a Boolean Function. IRE Transac-



- tions on EC Vol. EC-5 № 9. September 1956. 126-132.
- 67 E.W. Veitch: A Cart Method for Simplifying Truth Functions. Proc. Assoc. Computing Machinery, 1952. 127-133.
- 68 Е.К.Бойшвилло: Метод упрощения форм выражения функций истинности. Научные доклады высшей школы (Философические науки) 2. (1958) 120-135.
- 69 J.N. Warfield: A Note on the Reductions of Switching Functions. IRE Transactions on EC Vol. EC-7. № 6. June 1958. 180-181.
- 70 G. Warlitz: Über eine Schaltungsalgebra für NOR und NAND Elektronische Datenverarbeitung, 1967. H.2. 65-73.
- 71 Ю.И.Журавлев: Оценка для числа тупиковых ДНФ функций алгебры логики. Сибирский Мат. Журн. 1962. №5.



# TARTALOMJEGYZÉK

	Oldal
ELŐSZÓ . . . . .	3
BEVEZETÉS . . . . .	7
I. Fejezet	
ALAPFOGALMAK ÉS DEFINÍCIÓK . . . . .	13
II. Fejezet	
BOOLE-FÜGGVÉNYEK MINIMÁLIS VAGY NEM REDUNDÁNS DISZJUNKTIV NORMÁLFORMÁINAK ELŐÁLLÍTÁSA . . . . .	41
II.1. Boole-függvények minimális diszjunktiv normál- formájának meghatározására szolgáló fontosabb módszerek áttekintése . . . . .	41
II.2. Egyes algoritmusok módosításai . . . . .	60
II.3. Gráf-módszer . . . . .	72
III. Fejezet	
BOOLE-FÜGGVÉNYEK SZINTÉZISE KÜLÖNBÖZŐ BÁZISOKBAN BIZONYOS KORLÁTOZÁSOK FIGYELEMBEVÉTELÉVEL . . . . .	97
III.1. Ismert módszerek . . . . .	99
III.2. A T táblázat . . . . .	103
III.3. $\Phi$ -Boole-függvények szintézise $\{\wedge, \vee, \neg\}$ bázisban a határok felhasználásával . . . . .	109
III.4. Szintézis $\{\text{NAND}\}$ vagy $\{\text{NOR}\}$ bázisban . . . . .	112
III.5. Általános Boole-függvények szintézise a határok össze- hasonlítási módszerével $\{\wedge, \vee, \neg\}$ és $\{\text{NAND}\}$ bázisban . . .	116
FÜGGELÉK . . . . .	123
IRODALOMJEGYZÉK . . . . .	129



1070

1. The first part of the report is a general introduction to the subject of the study.

2. The second part of the report is a detailed description of the methods used in the study.

3. The third part of the report is a discussion of the results of the study.

4. The fourth part of the report is a conclusion and a list of references.

5. The fifth part of the report is a list of appendices.

6. The sixth part of the report is a list of figures and tables.

7. The seventh part of the report is a list of footnotes.

8. The eighth part of the report is a list of abbreviations.

9. The ninth part of the report is a list of symbols.

10. The tenth part of the report is a list of definitions.

11. The eleventh part of the report is a list of acknowledgments.

12. The twelfth part of the report is a list of references.

13. The thirteenth part of the report is a list of appendices.

14. The fourteenth part of the report is a list of figures and tables.





